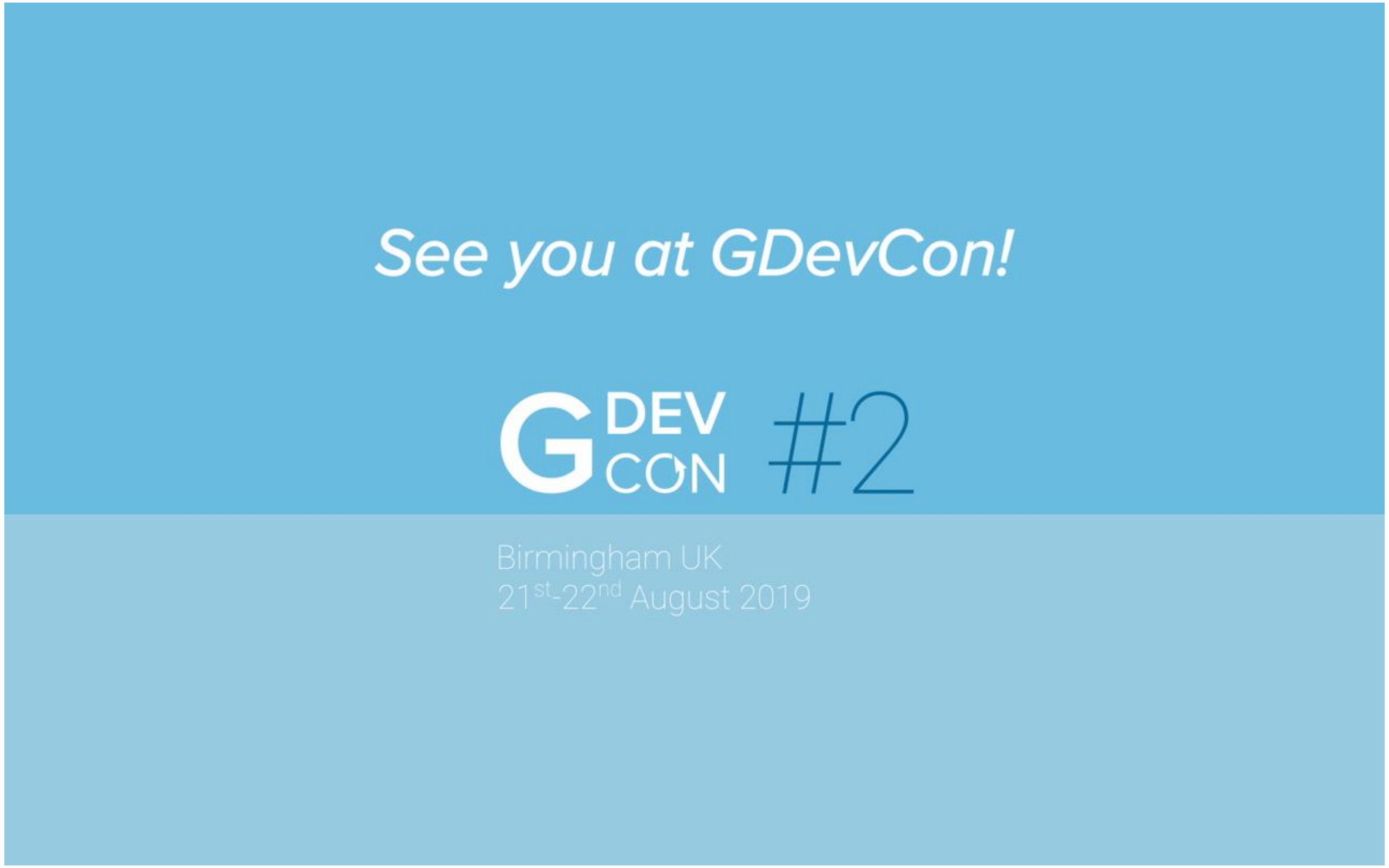
About the Session Author

- Allen C Smith has been using the NI platform to solve complex engineering challenges for over 26 years.
- He is a Certified LabVIEW Architect, Certified Embedded Developer, Certified TestStand Developer, and LabVIEW Champion.
- He has been an Alliance Partner, worked for Alliance Partners, and served as a Systems Engineer with National Instruments. He is once again an Alliance Partner, offering services as a consulting software architect and engineer.
- While at NI, he developed the software tool support and official training course for the Actor Framework. He remains an AF evangelist and active member of the AF community.

- Follow me at
  - linkedin.com/in/allencsmith
  - Allencsmith149.workfolio.com
  - justACS

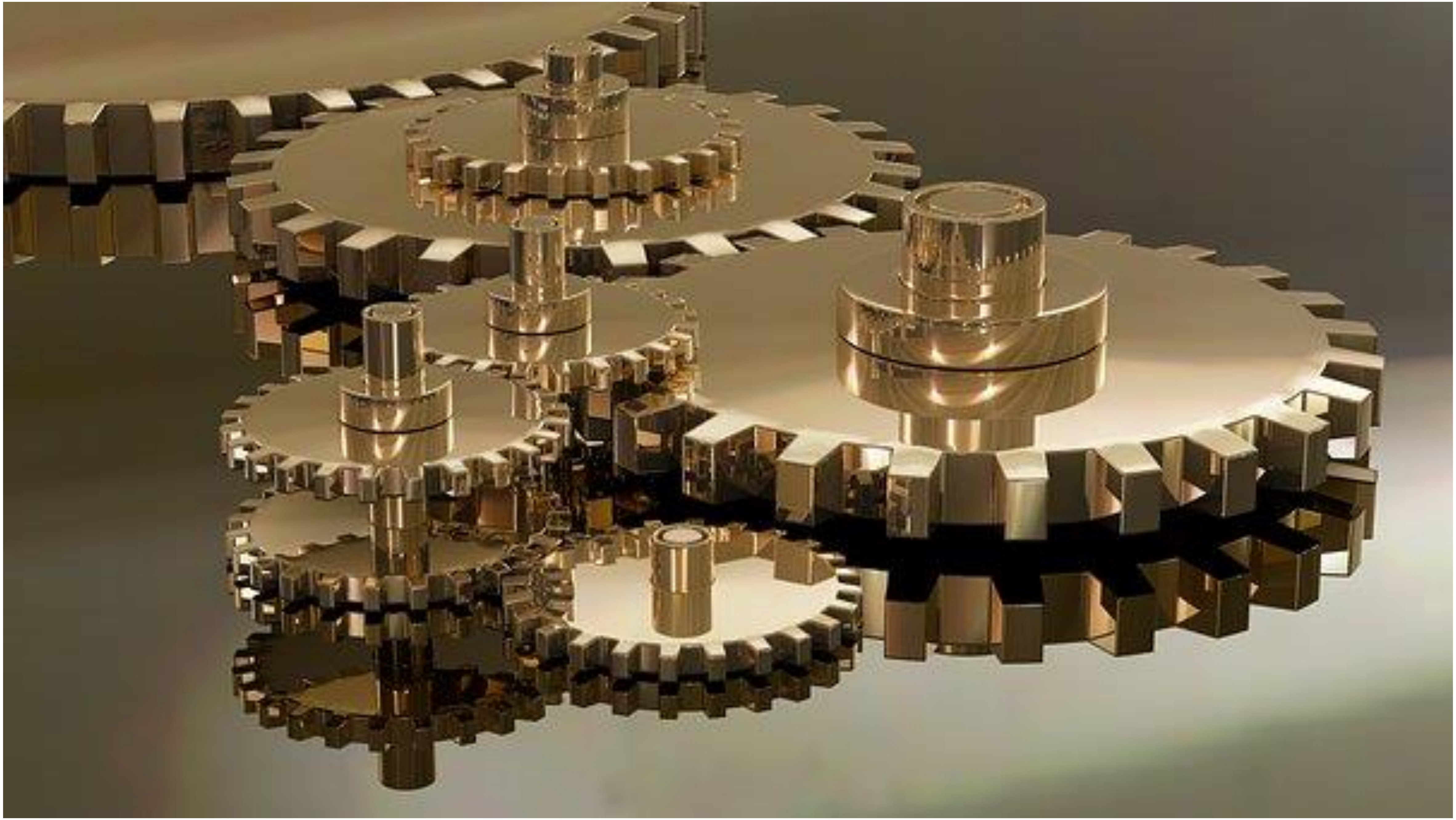
justACS is a proud sponsor of GDevCon!

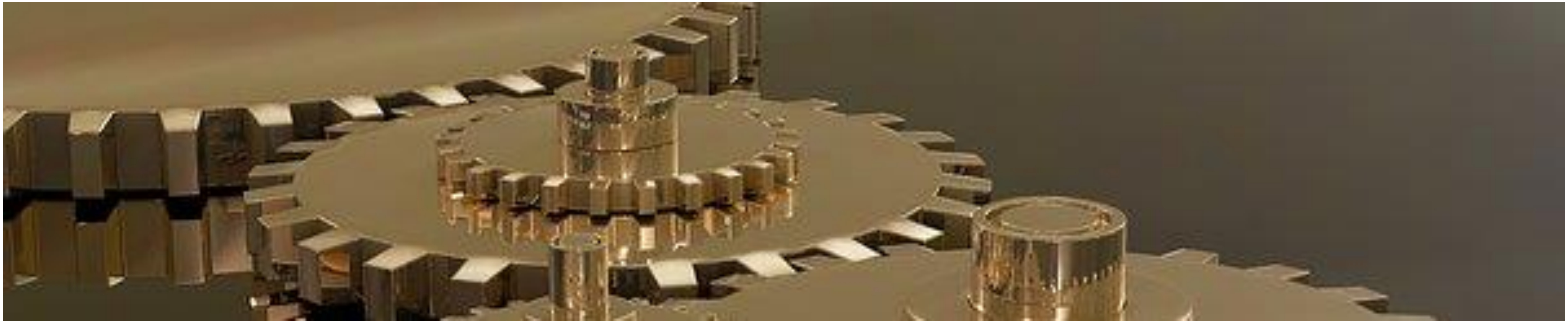


See you at GDevCon!

G DEV CON #2

Birmingham UK
21st-22nd August 2019





Test Driven Development in Actor Framework

Allen C Smith

CLA, CLED, CTD, CPI, LabVIEW Champion

WELCOME TO THE PARTY PAL!

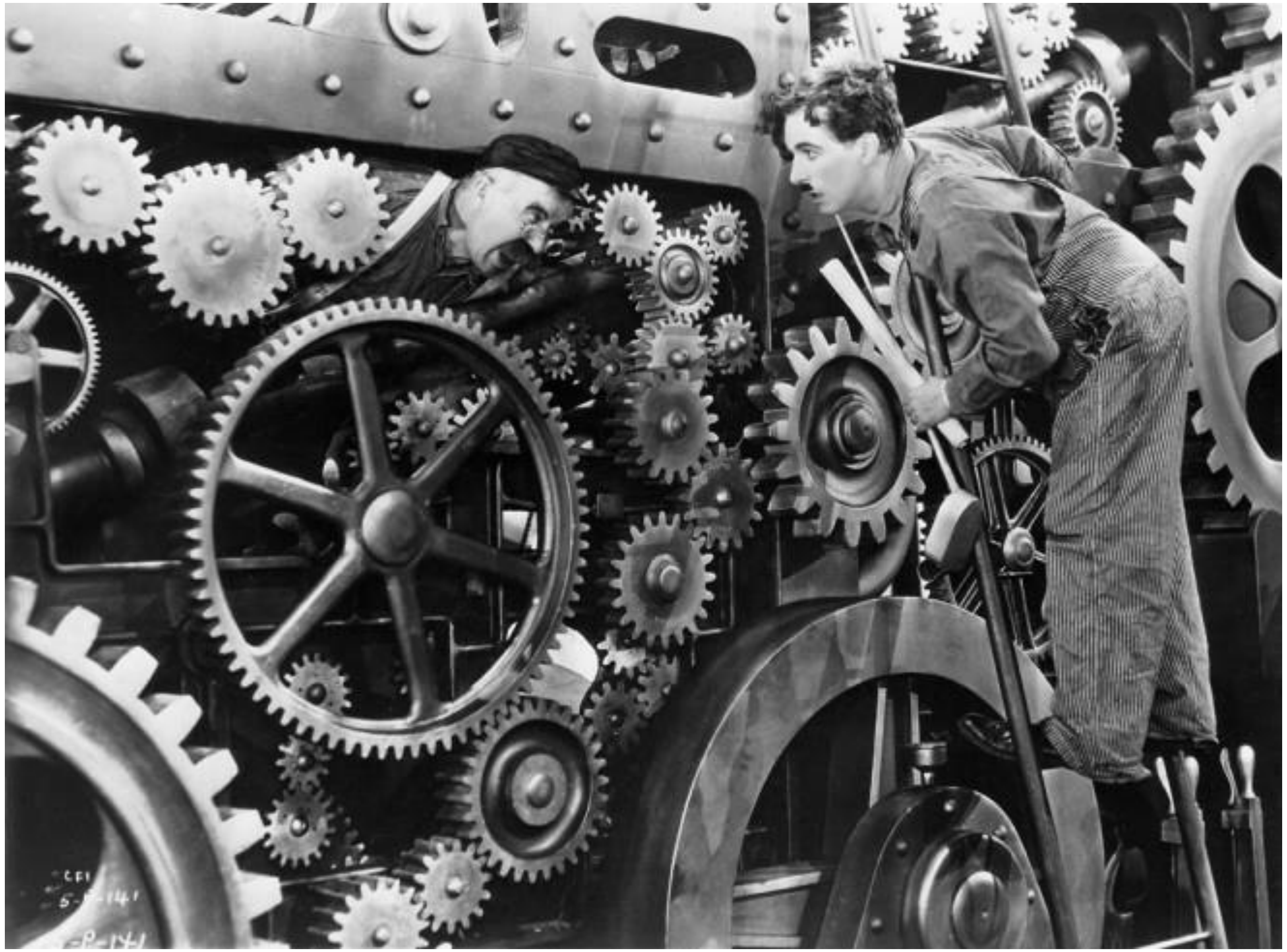
Agenda

- Actor Framework – A Testing Challenge
- Isolating Actors
- Example – Testing a Simple Feature
  - Actor sends a message
  - Actor changes state
- A Cost Analysis

Agenda

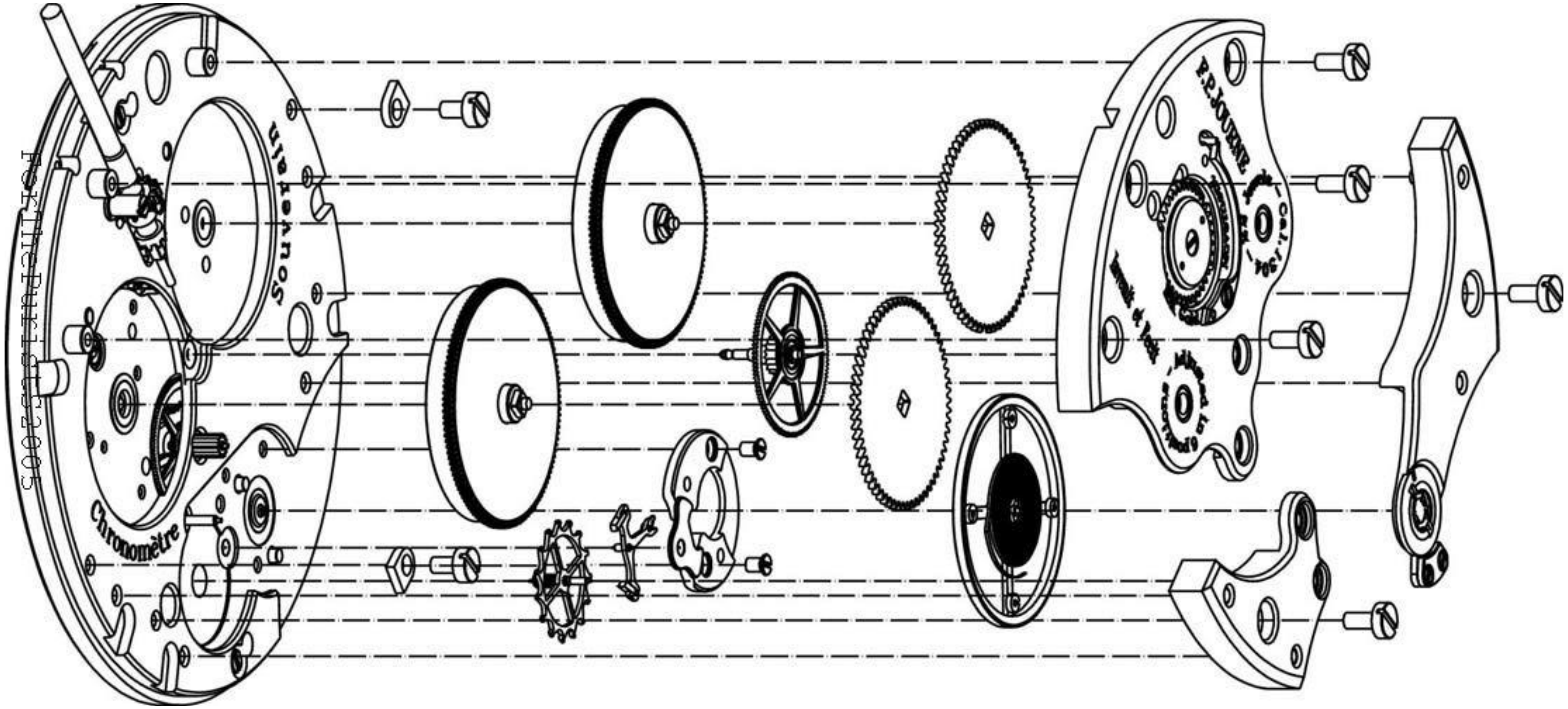
- Actor Framework – A Testing Challenge
- Isolating Actors
- Example – Testing a Simple Feature
  - Actor sends a message
  - Actor changes state
- A Cost Analysis

Our Challenge: Go from this…

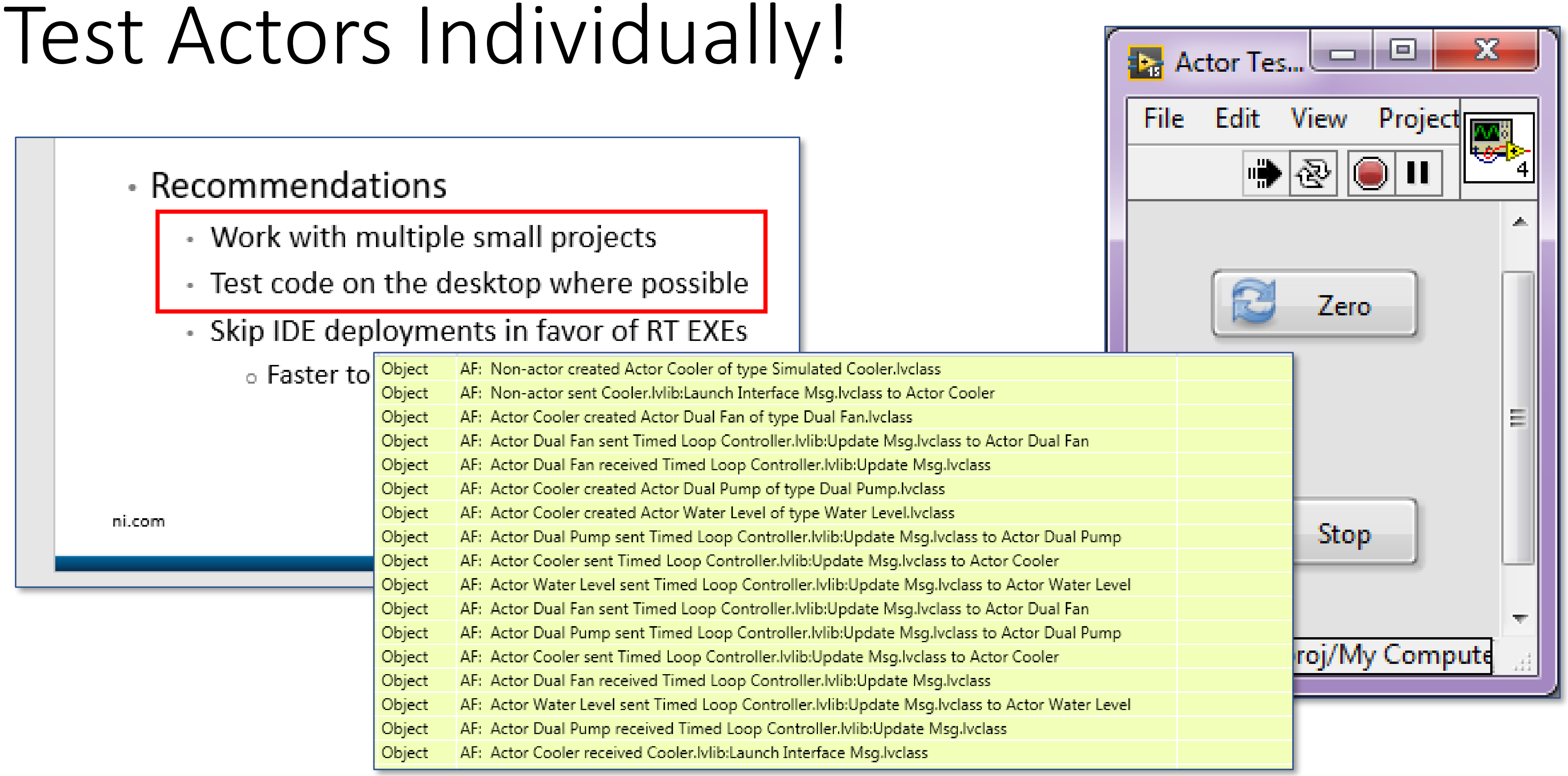


Seen at chucksblog.typepad.com

… to this


ForThePurists2005
Souverain
Chronomètre
F.P.JOURNE
Cal.1304
Invenit et Fecit

Test Actors Individually!

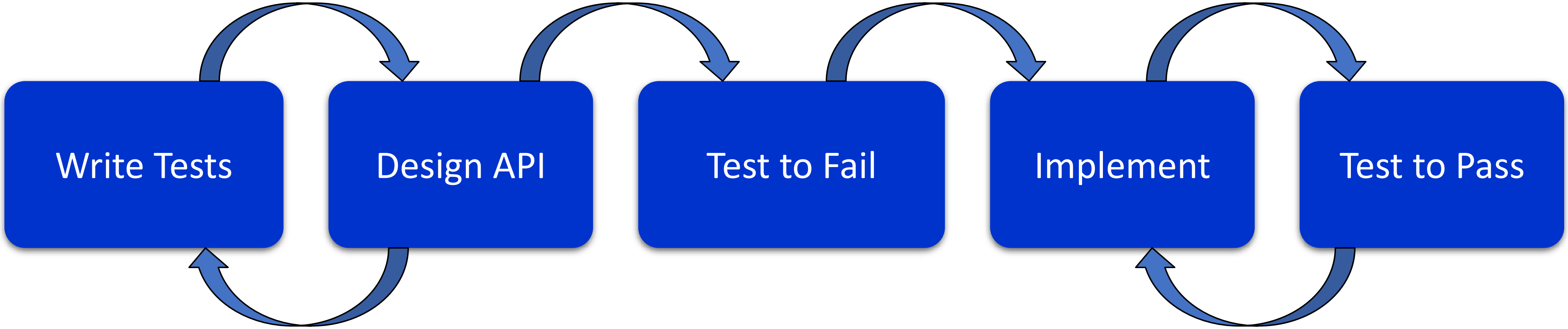


- Recommendations
  - Work with multiple small projects
  - Test code on the desktop where possible
  - Skip IDE deployments in favor of RT EXEs
    - Faster to

ni.com

```
Object    AF:  Non-actor created Actor Cooler of type Simulated Cooler.lvclass
Object    AF:  Non-actor sent Cooler.lvlib:Launch Interface Msg.lvclass to Actor Cooler
Object    AF:  Actor Cooler created Actor Dual Fan of type Dual Fan.lvclass
Object    AF:  Actor Dual Fan sent Timed Loop Controller.lvlib:Update Msg.lvclass to Actor Dual Fan
Object    AF:  Actor Dual Fan received Timed Loop Controller.lvlib:Update Msg.lvclass
Object    AF:  Actor Cooler created Actor Dual Pump of type Dual Pump.lvclass
Object    AF:  Actor Cooler created Actor Water Level of type Water Level.lvclass
Object    AF:  Actor Dual Pump sent Timed Loop Controller.lvlib:Update Msg.lvclass to Actor Dual Pump
Object    AF:  Actor Cooler sent Timed Loop Controller.lvlib:Update Msg.lvclass to Actor Cooler
Object    AF:  Actor Water Level sent Timed Loop Controller.lvlib:Update Msg.lvclass to Actor Water Level
Object    AF:  Actor Dual Fan sent Timed Loop Controller.lvlib:Update Msg.lvclass to Actor Dual Fan
Object    AF:  Actor Dual Pump sent Timed Loop Controller.lvlib:Update Msg.lvclass to Actor Dual Pump
Object    AF:  Actor Cooler sent Timed Loop Controller.lvlib:Update Msg.lvclass to Actor Cooler
Object    AF:  Actor Dual Fan received Timed Loop Controller.lvlib:Update Msg.lvclass
Object    AF:  Actor Water Level sent Timed Loop Controller.lvlib:Update Msg.lvclass to Actor Water Level
Object    AF:  Actor Dual Pump received Timed Loop Controller.lvlib:Update Msg.lvclass
Object    AF:  Actor Cooler received Cooler.lvlib:Launch Interface Msg.lvclass
```

Test Driven Development: Work Flow



Write Tests

Design API

Test to Fail

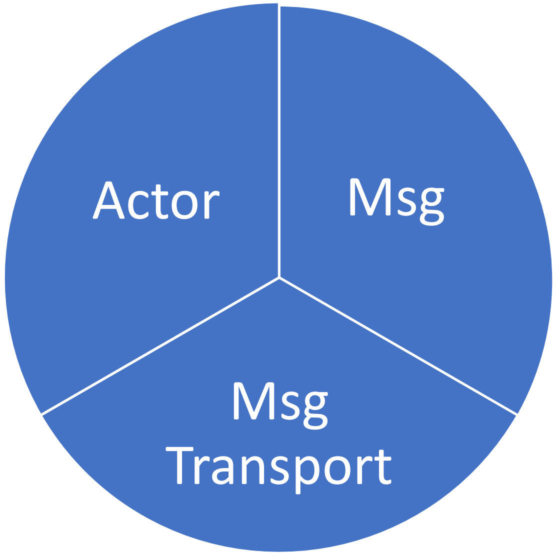
Implement

Test to Pass

Agenda

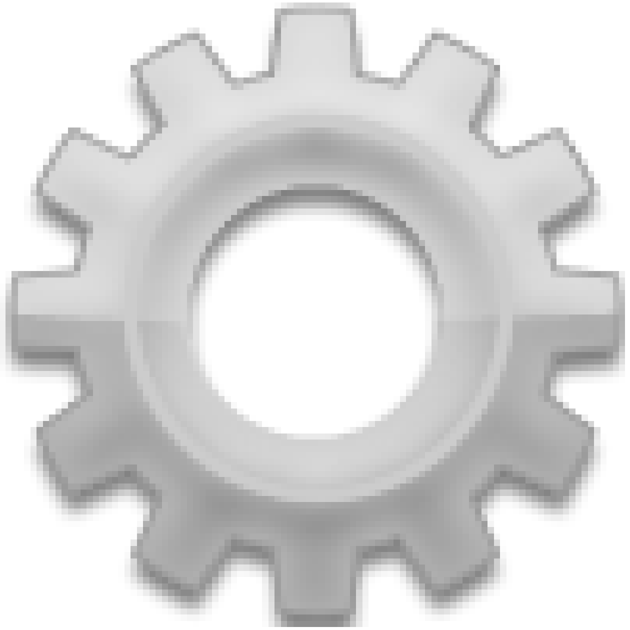
- Actor Framework – A Testing Challenge
- Isolating Actors
- Example – Testing a Simple Feature
  - Actor sends a message
  - Actor changes state
- A Cost Analysis

Elements of the Actor Model



- Actor
  An encapsulation of data, behaviors, and threads
- Message
  A unit of information passed between actors
- Message Transport
  A mechanism to move messages to the destination

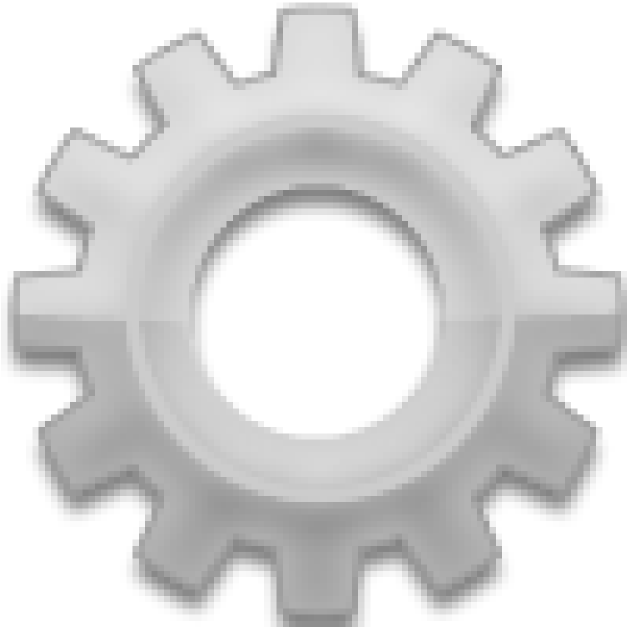
Elements of the Actor Model



An Actor can:

- Send a message
- Launch an actor
- Change how it will respond to the next message it receives

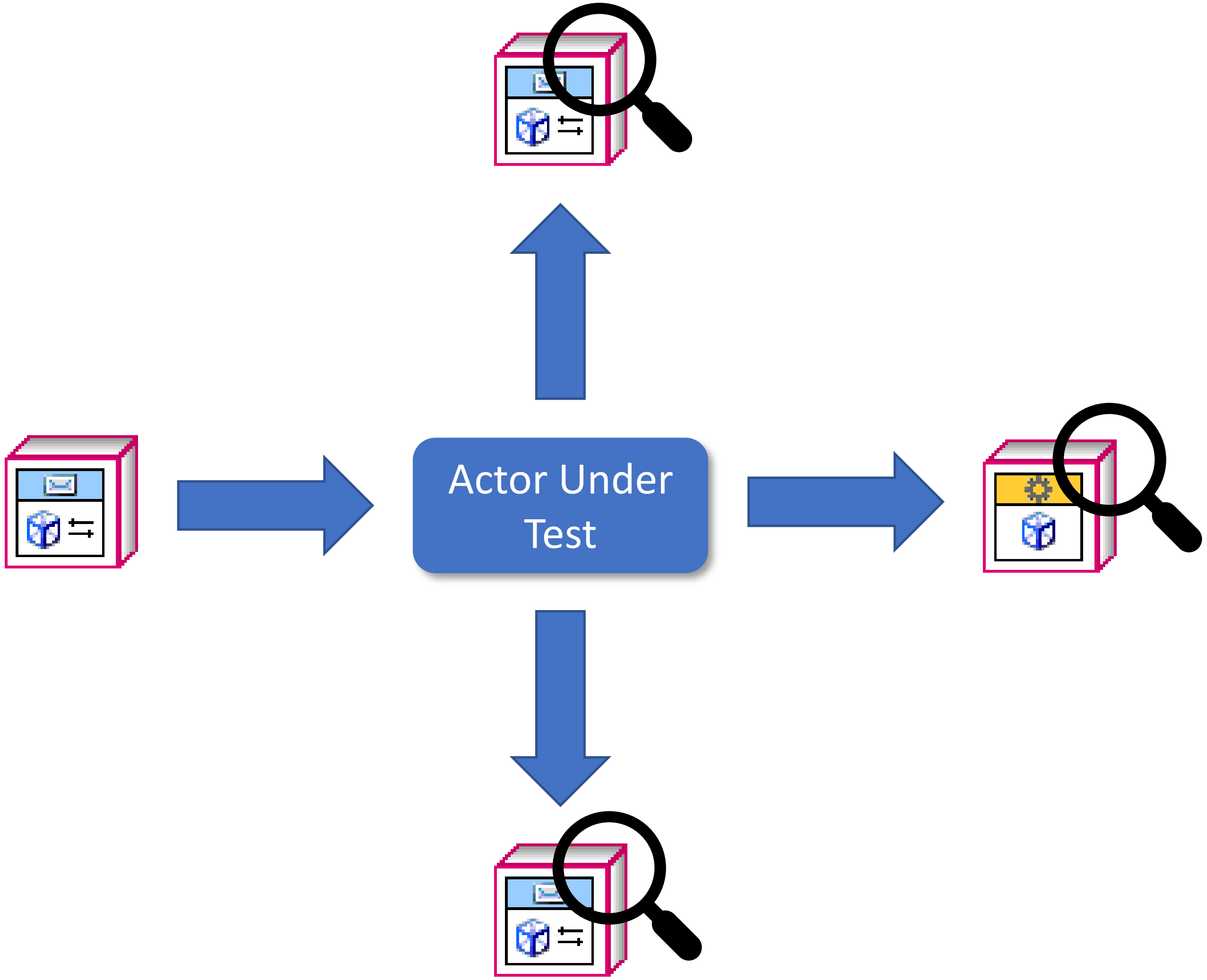
Elements of the Actor Model



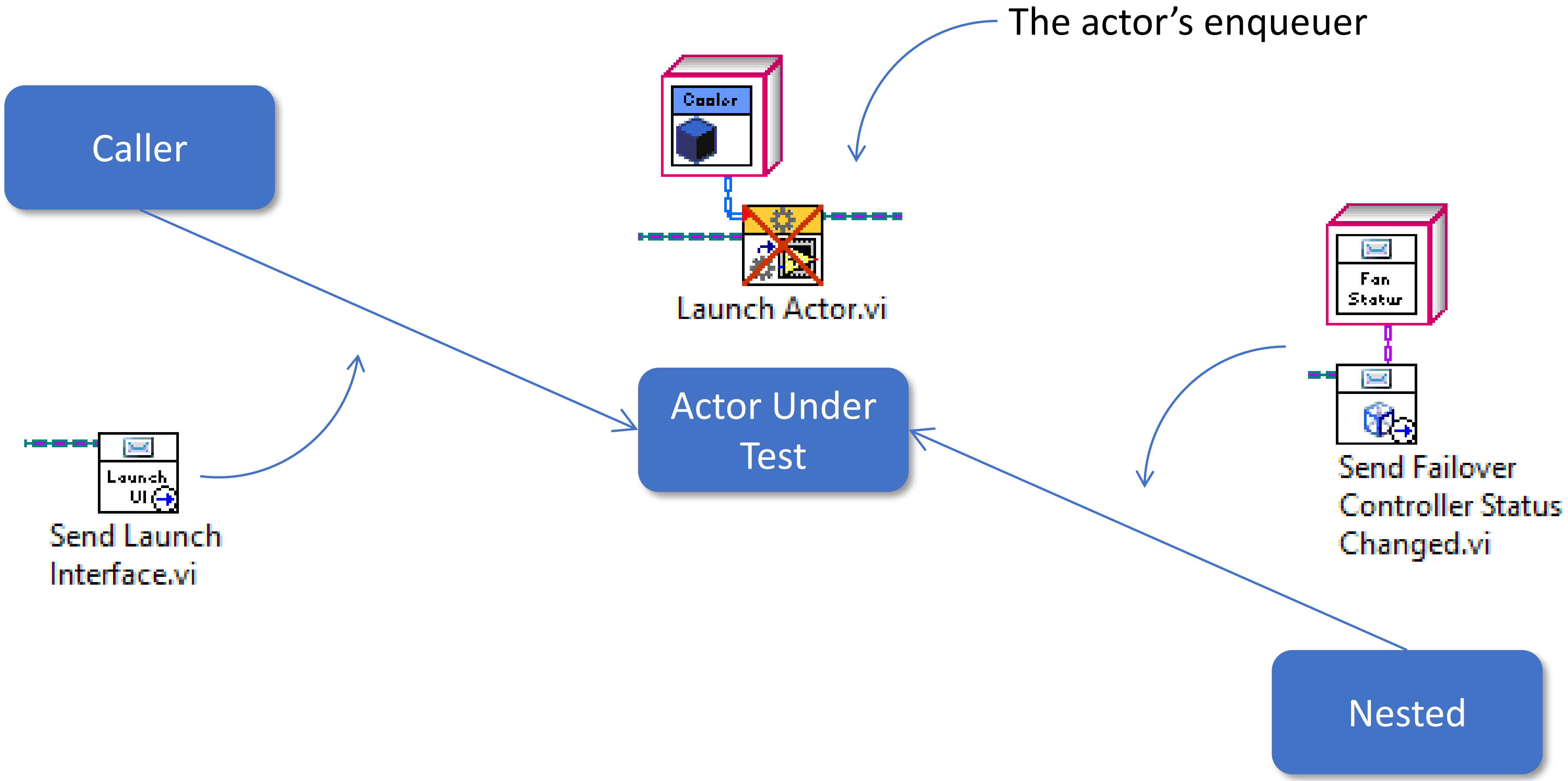
An Actor can:

- Send a message
- ~~Launch an actor~~
- Change how it will respond to the next message it receives

How to Test an Actor



Send a message to the actor…



The actor's enqueuer

Caller

Launch Actor.vi

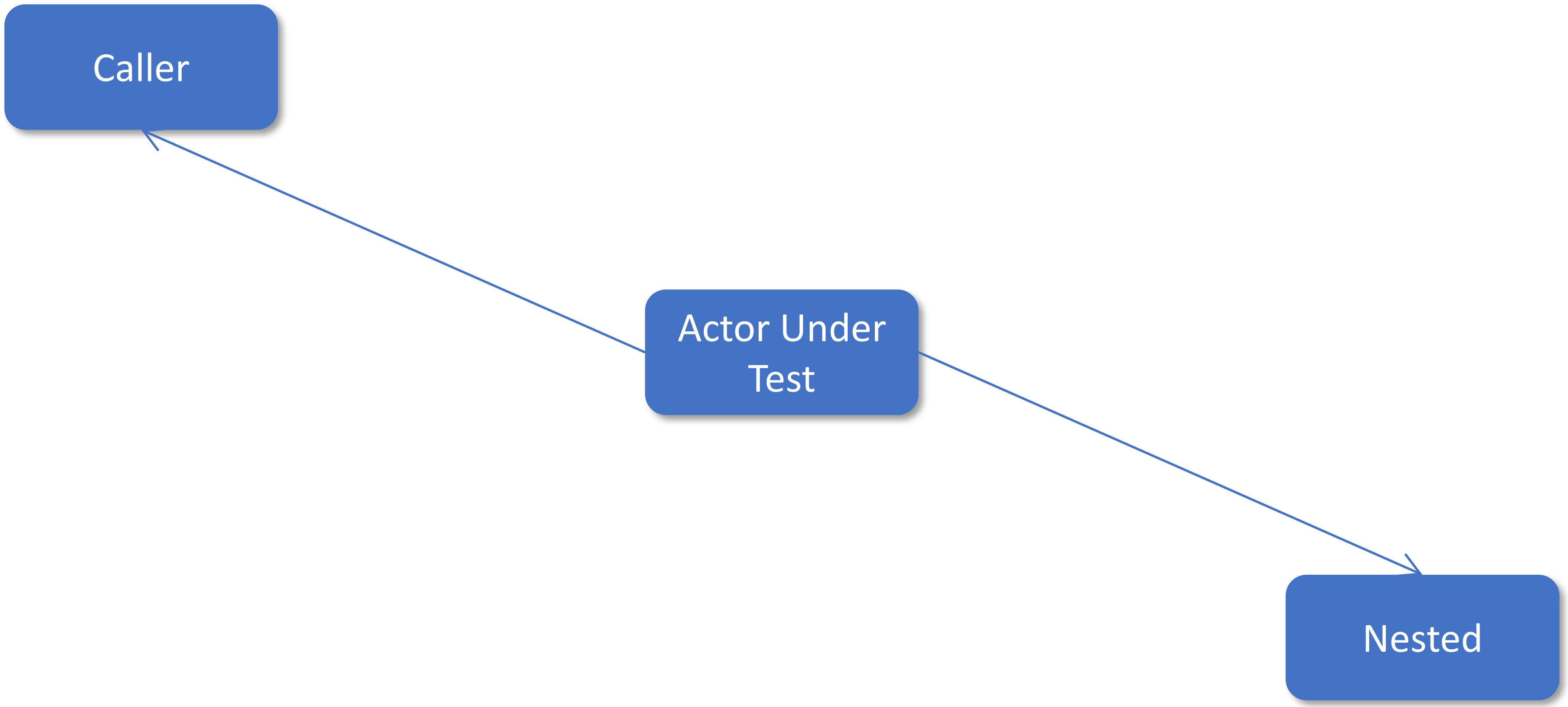
Actor Under Test

Send Launch Interface.vi

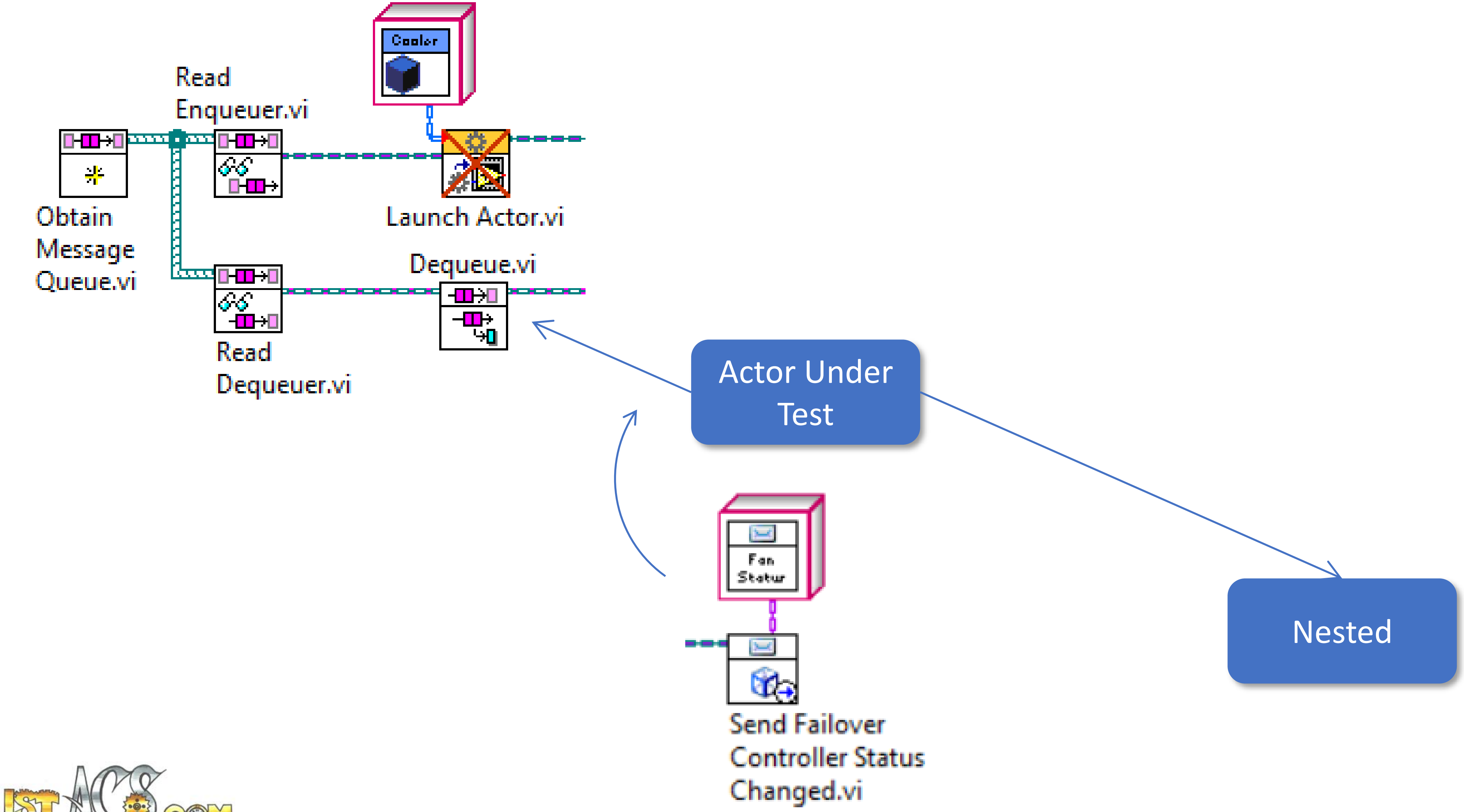
Send Failover Controller Status Changed.vi

Nested

Receiving a message from the actor…



Receiving a message for the nested



Read Enqueuer.vi

Obtain Message Queue.vi

Launch Actor.vi

Dequeue.vi

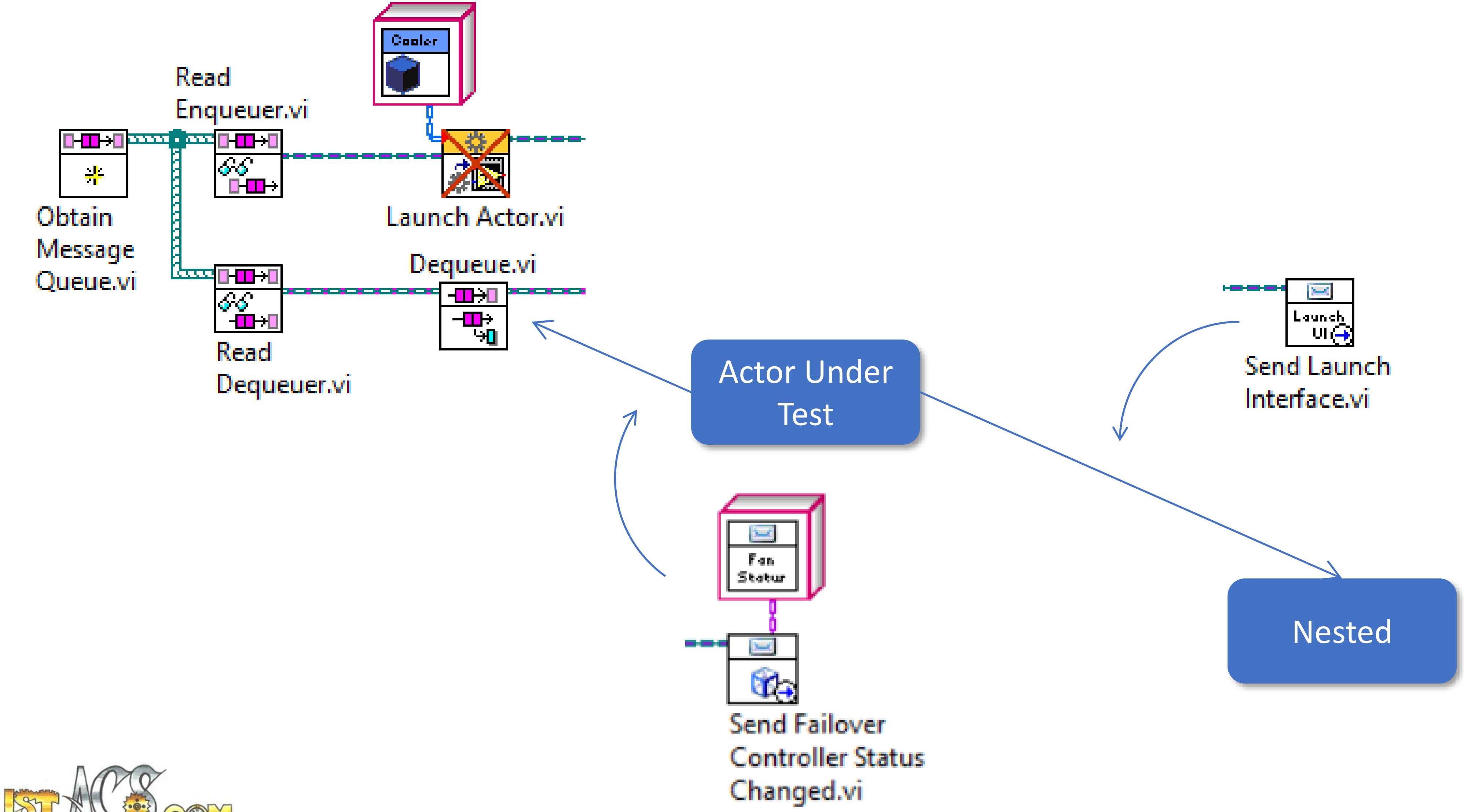
Read Dequeuer.vi

Actor Under Test

Nested

Send Failover Controller Status Changed.vi

Receiving a message for the nested



Read Enqueuer.vi

Obtain Message Queue.vi

Launch Actor.vi

Dequeue.vi

Read Dequeuer.vi

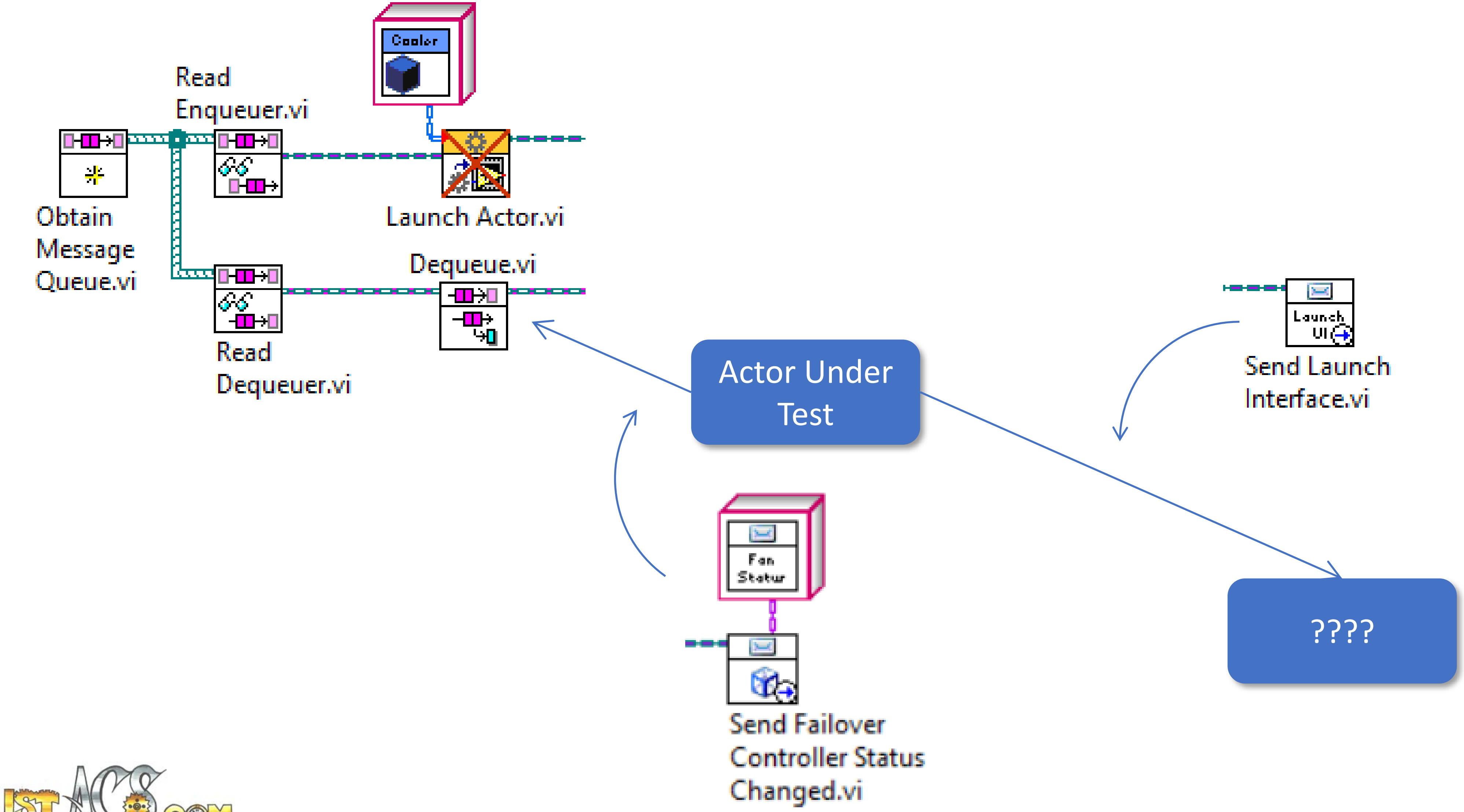
Actor Under Test

Send Launch Interface.vi

Nested

Send Failover Controller Status Changed.vi

Receiving a message for the nested



Read Enqueuer.vi

Obtain Message Queue.vi

Launch Actor.vi

Dequeue.vi

Read Dequeuer.vi

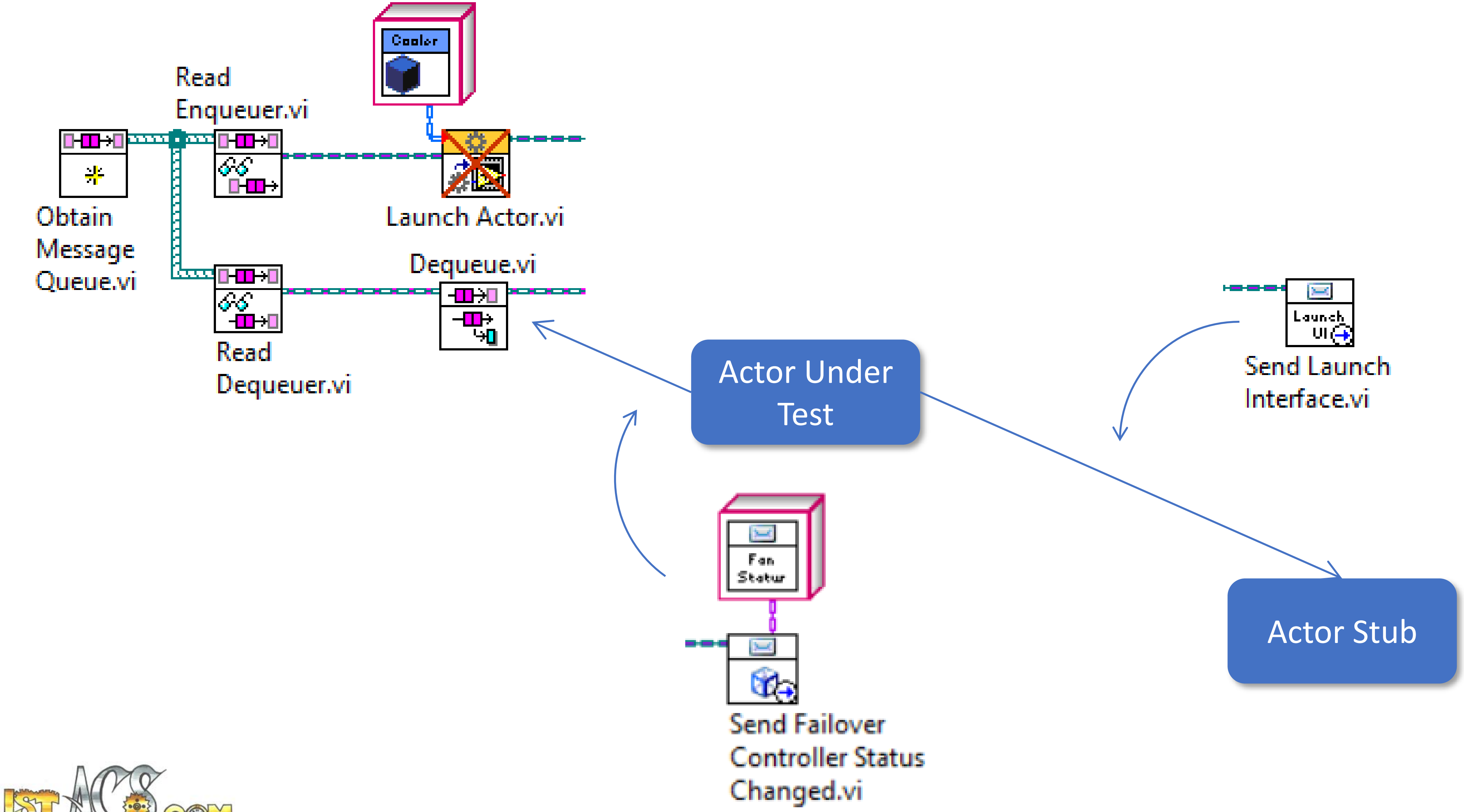
Actor Under Test

Send Launch Interface.vi

????

Send Failover Controller Status Changed.vi

Receiving a message for the nested



Read Enqueuer.vi

Obtain Message Queue.vi

Launch Actor.vi

Dequeue.vi

Read Dequeuer.vi

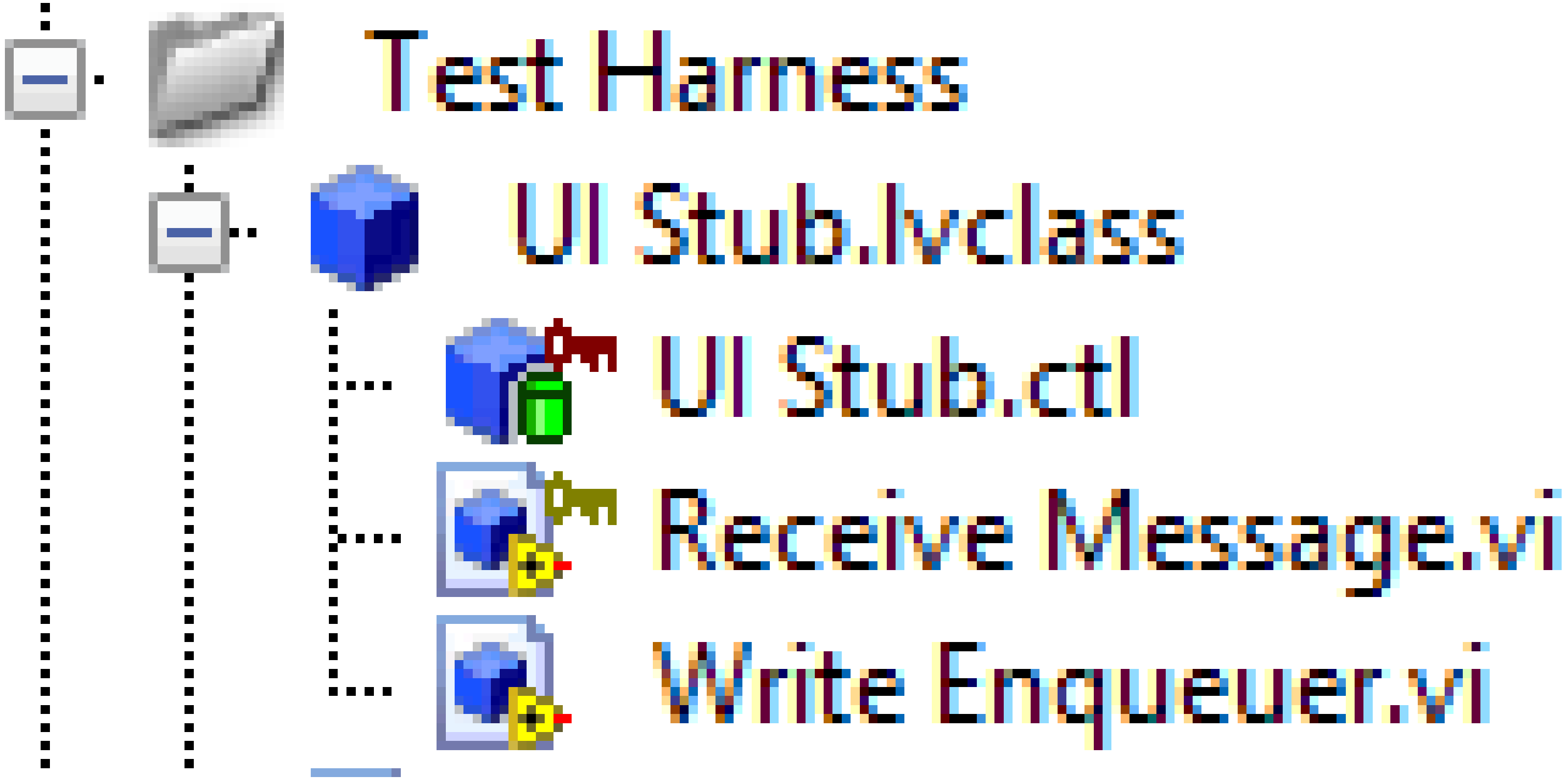
Actor Under Test

Send Launch Interface.vi

Actor Stub

Send Failover Controller Status Changed.vi

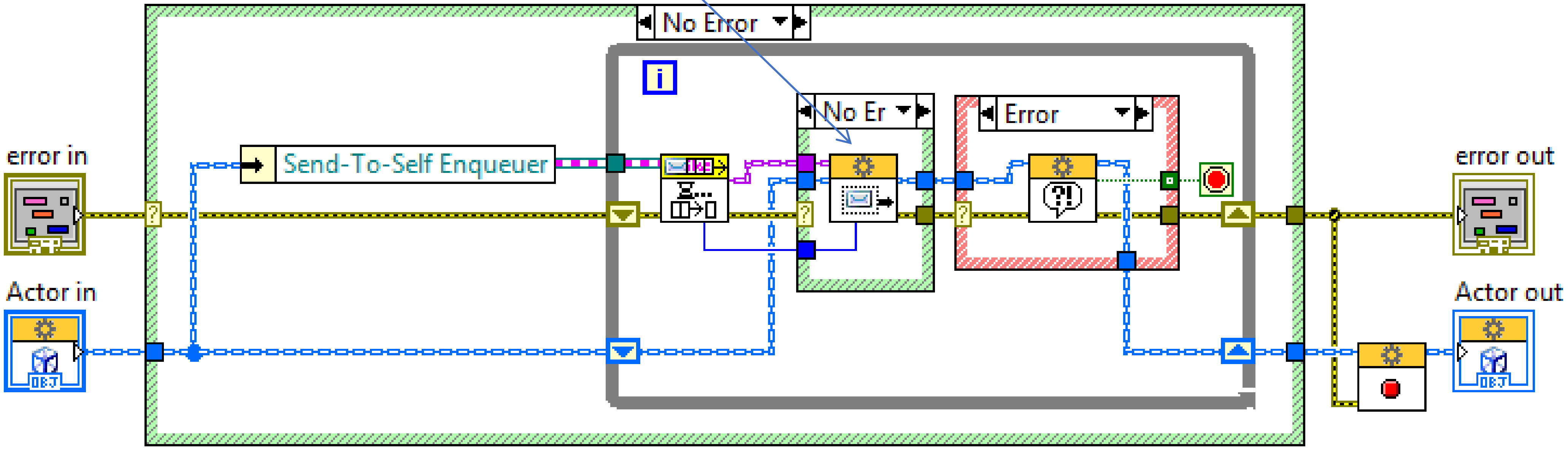
Actor Stubs



- Test Harness
  - UI Stub.lvclass
    - UI Stub.ctl
    - Receive Message.vi
    - Write Enqueuer.vi

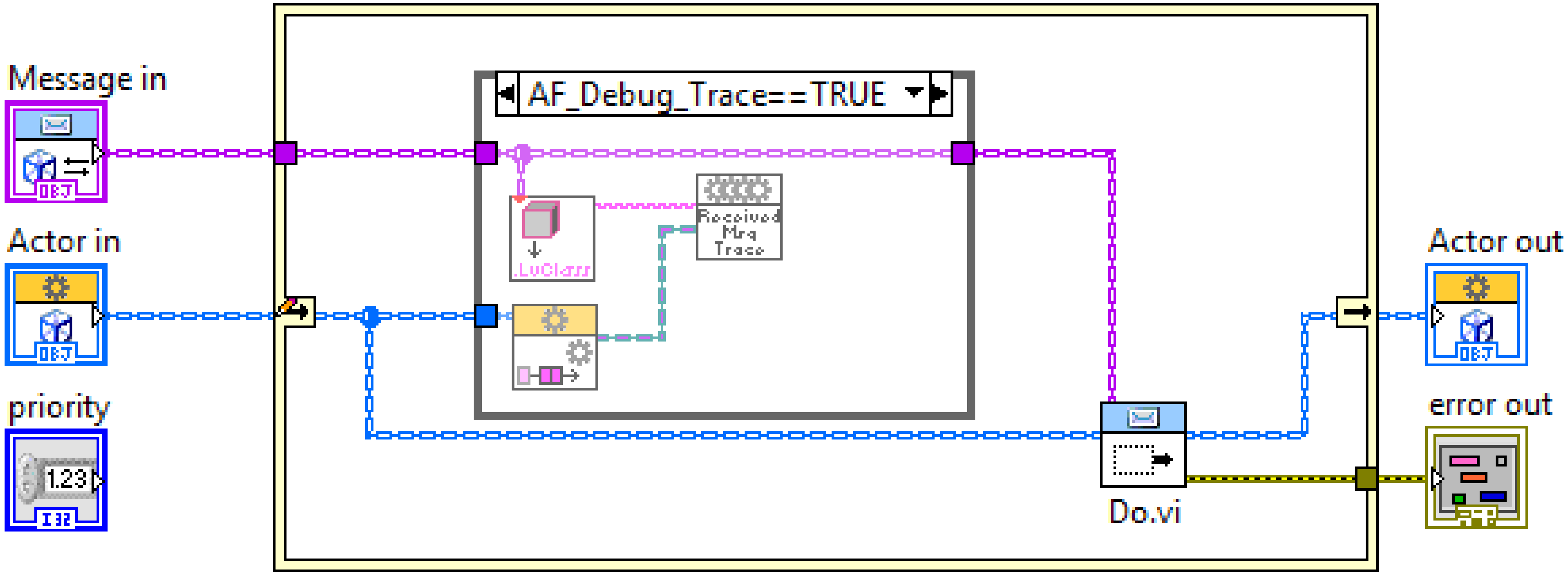
Actor Stubs – Override Receive Message.vi

Receive Message.vi

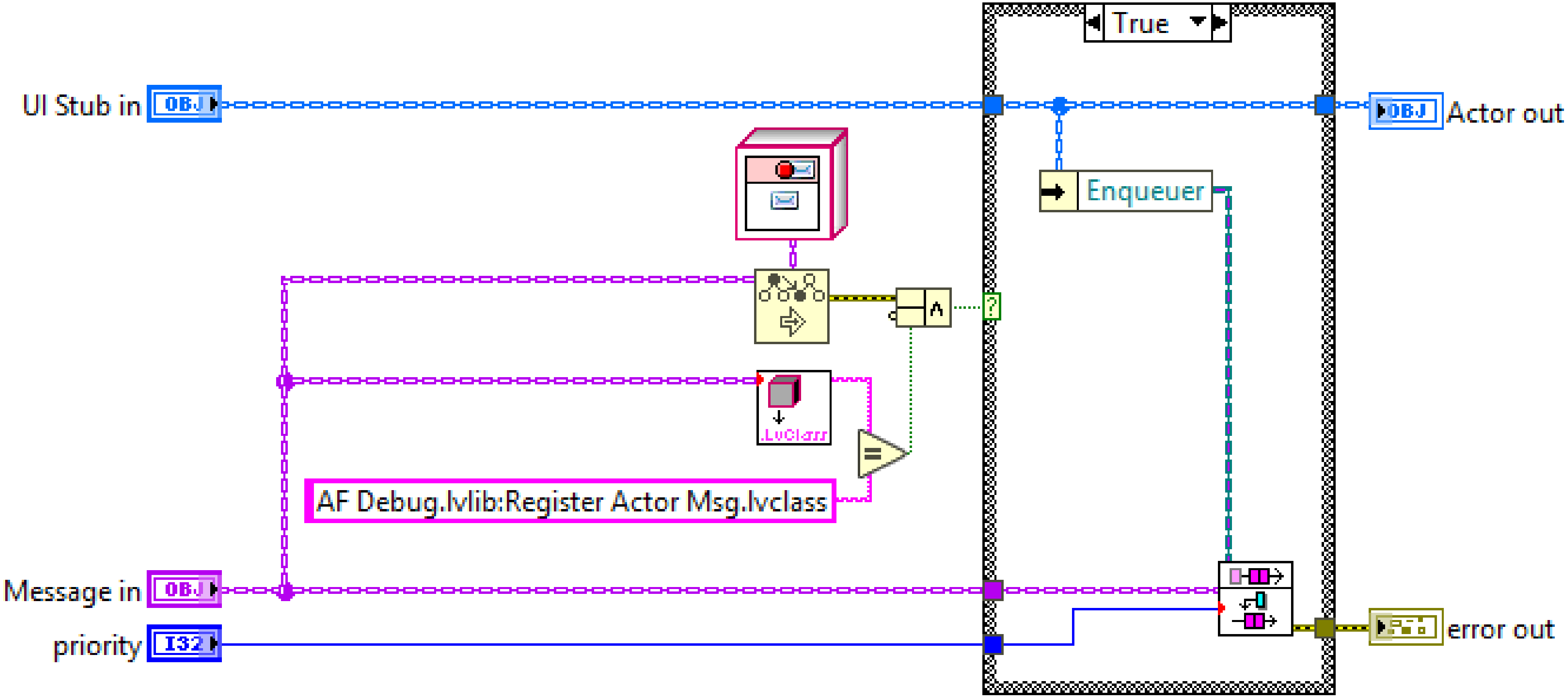


Actor:Actor Core

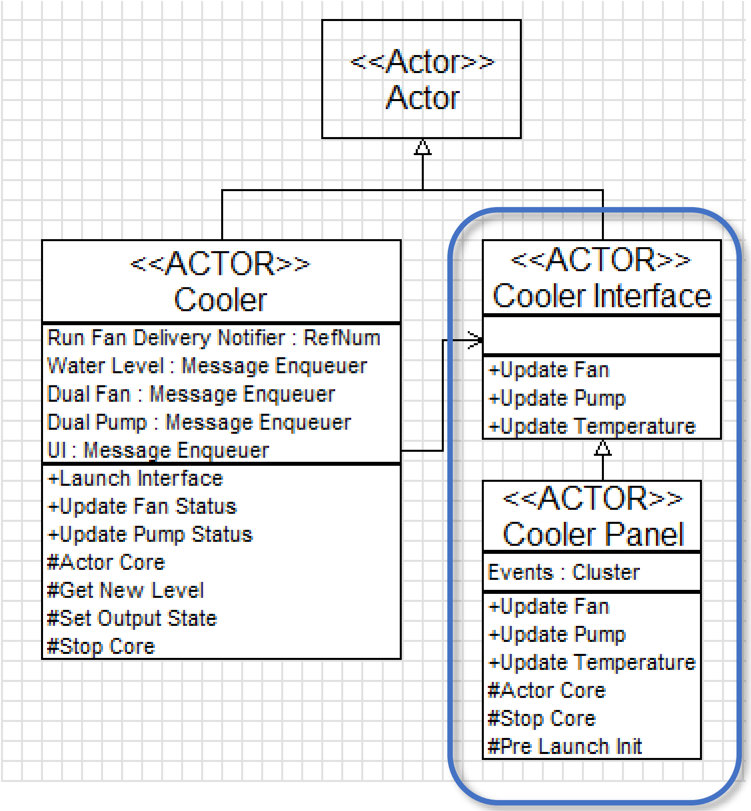
Actor Stubs – Override Receive Message.vi



Actor Stubs – Override Receive Message.vi



Interfaces/Low Coupling



<<Actor>>
Actor

<<ACTOR>>
Cooler

Run Fan Delivery Notifier : RefNum
Water Level : Message Enqueuer
Dual Fan : Message Enqueuer
Dual Pump : Message Enqueuer
UI : Message Enqueuer

+Launch Interface
+Update Fan Status
+Update Pump Status
#Actor Core
#Get New Level
#Set Output State
#Stop Core

<<ACTOR>>
Cooler Interface

+Update Fan
+Update Pump
+Update Temperature

<<ACTOR>>
Cooler Panel

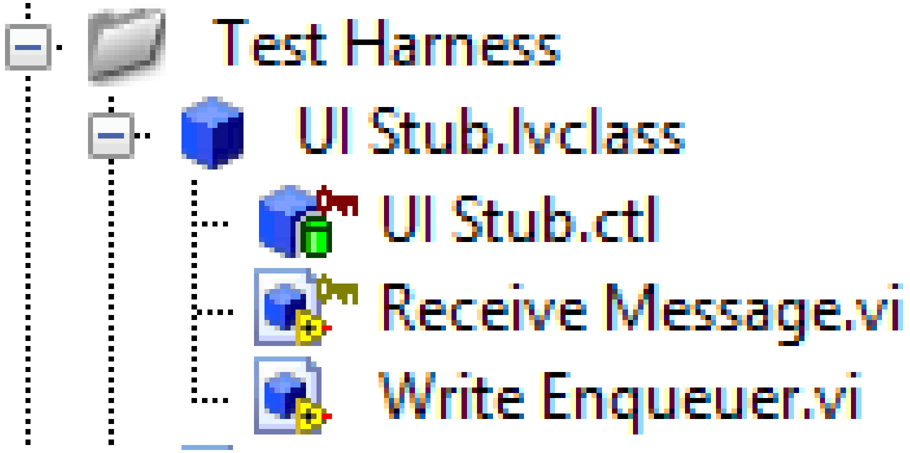
Events : Cluster

+Update Fan
+Update Pump
+Update Temperature
#Actor Core
#Stop Core
#Pre Launch Init

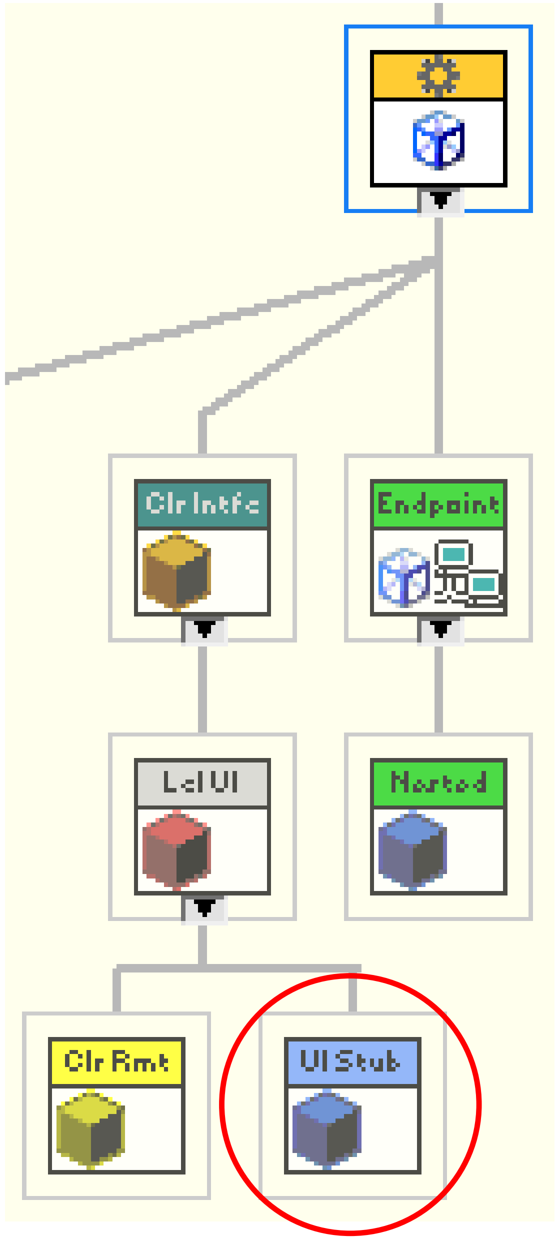
Abstract methods

Concrete Implementations

Dependency Injection



Test Harness
UI Stub.lvclass
UI Stub.ctl
Receive Message.vi
Write Enqueuer.vi



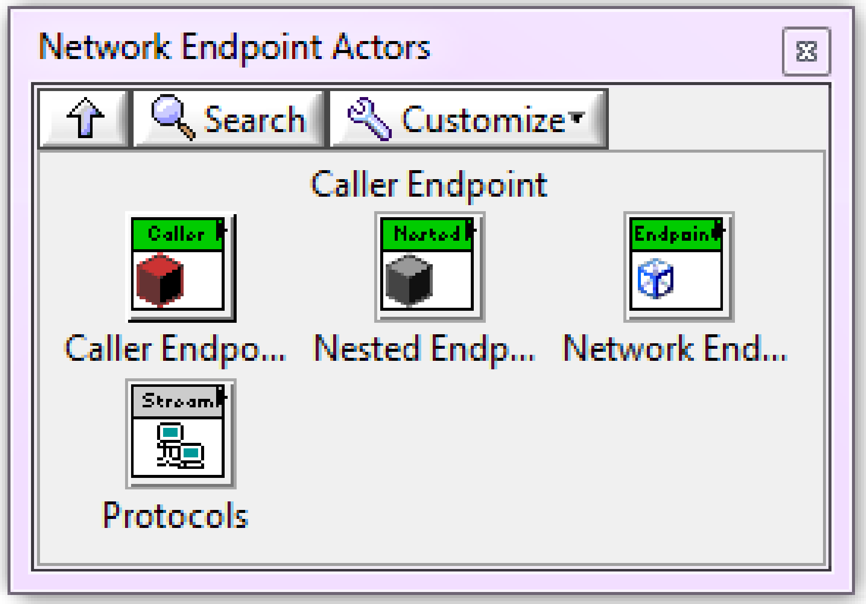
Agenda

- Actor Framework – A Testing Challenge
- Isolating Actors
- Example – Testing a Simple Feature
  - Actor sends a message
  - Actor changes state
- A Cost Analysis

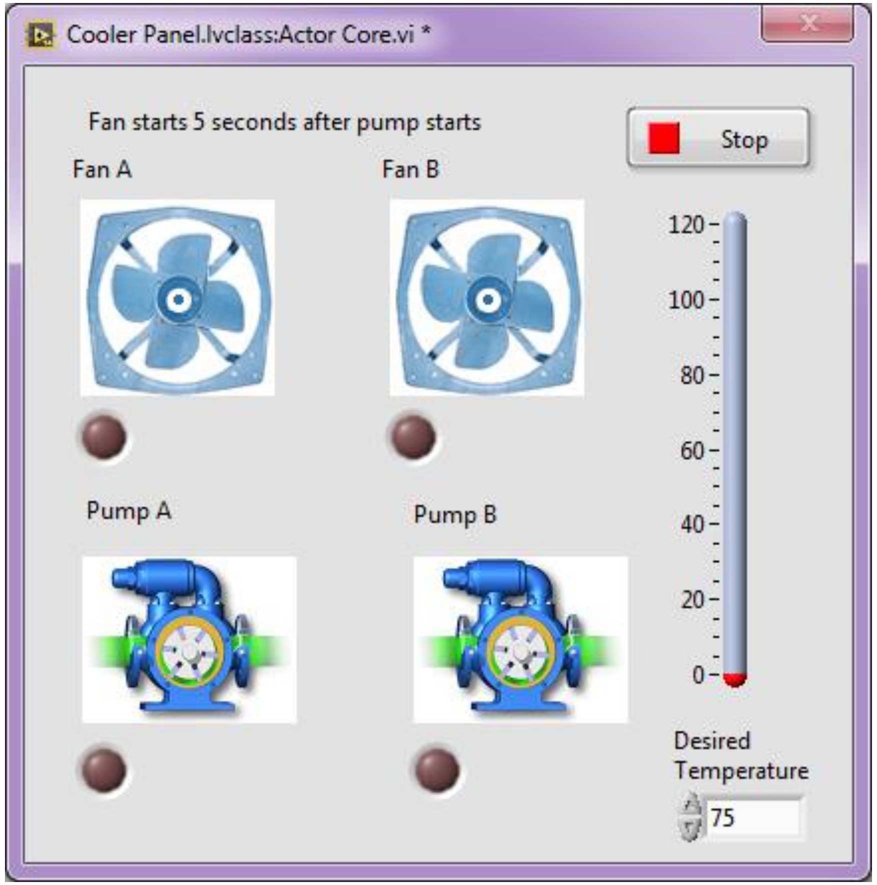
Better for Classes

Our Actors

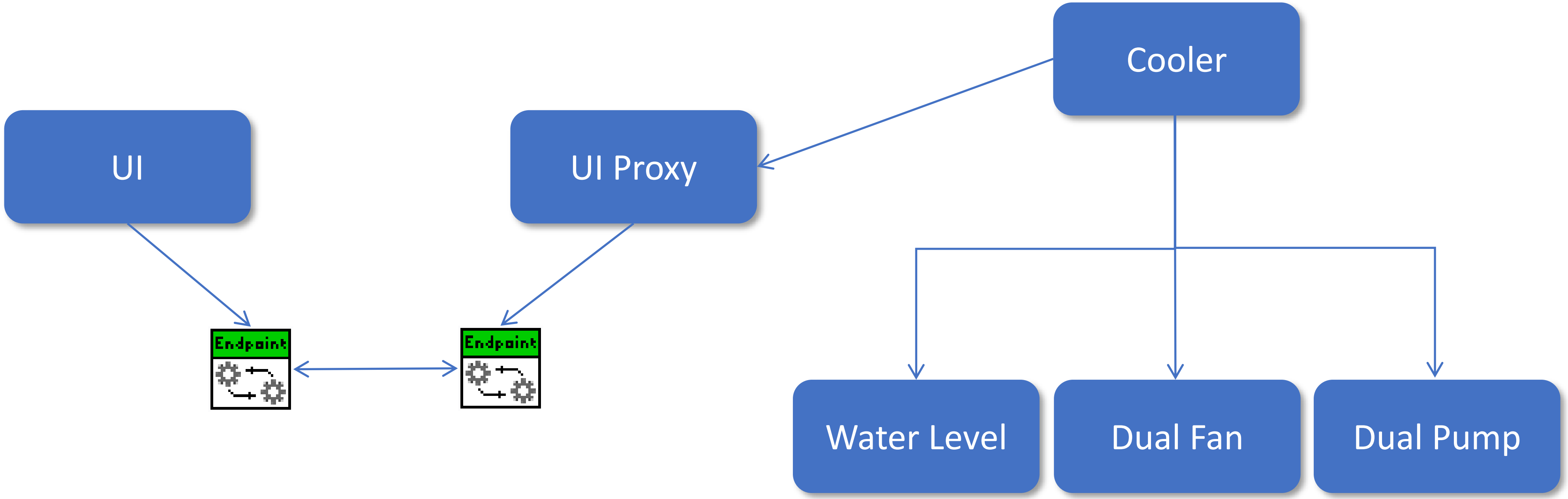
Network Endpoint Actors



Evaporative Cooler (Course Project)


Cooler Panel.lvclass:Actor Core.vi *
Fan starts 5 seconds after pump starts
Stop
Fan A
Fan B
Pump A
Pump B
120
100
80
60
40
20
0
Desired Temperature
75

System Diagram



UI

UI Proxy

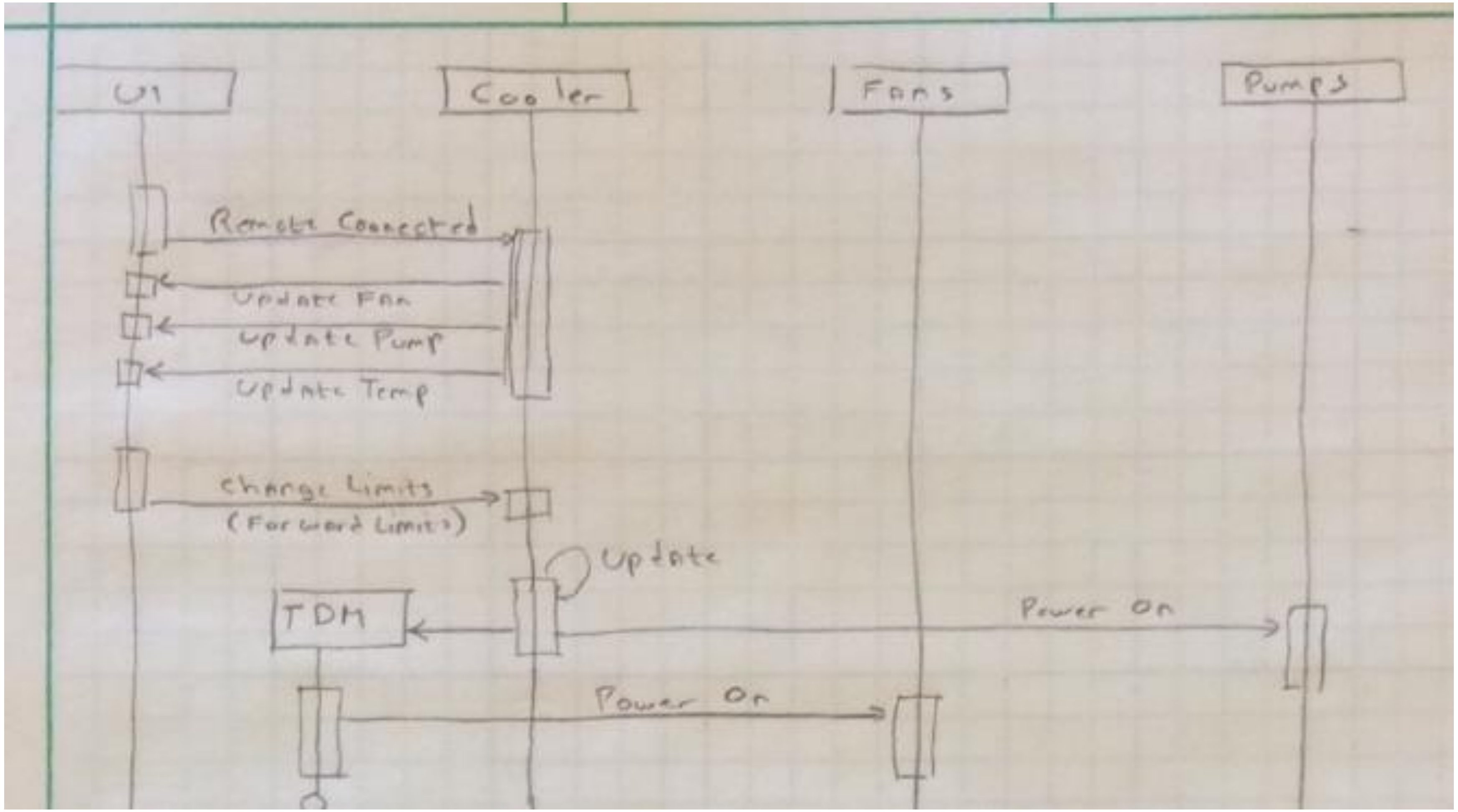
Cooler

Water Level

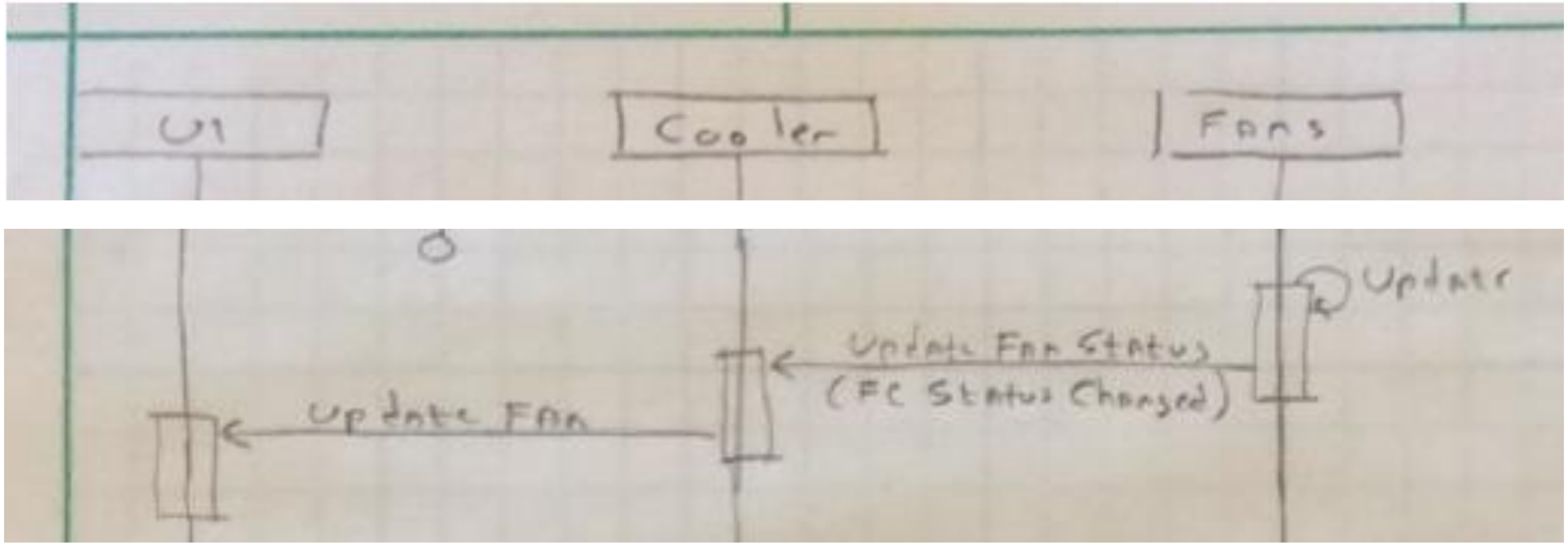
Dual Fan

Dual Pump

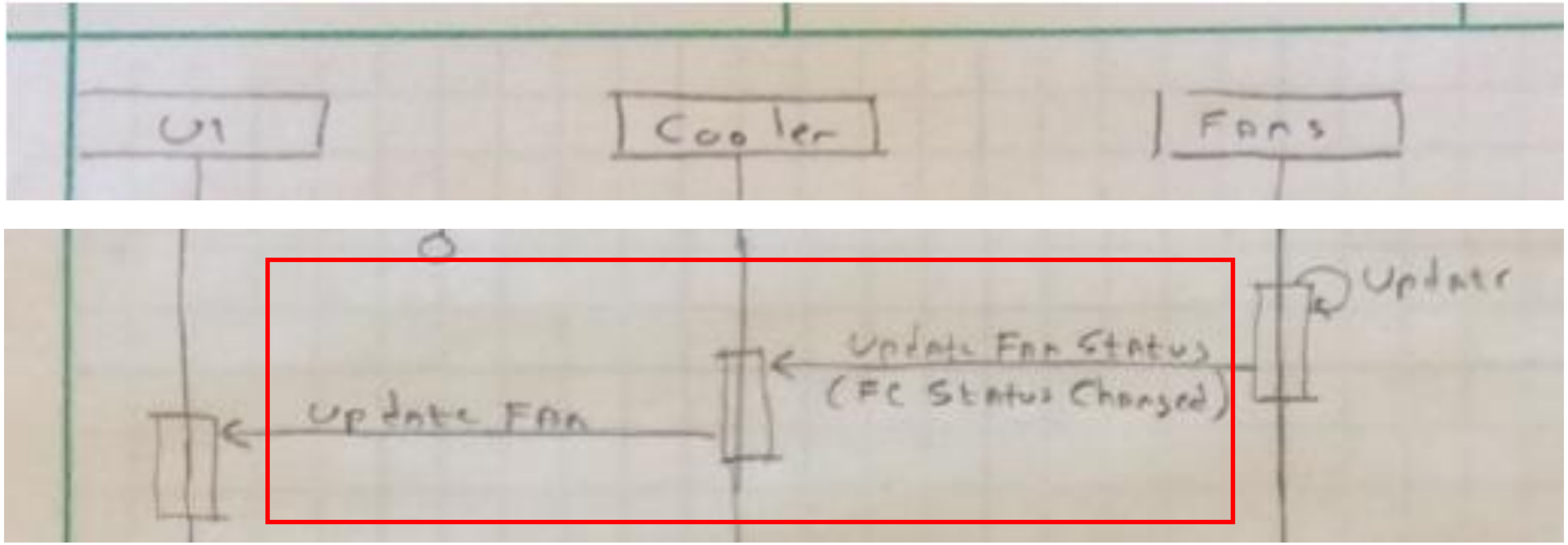
The Sequence Diagram



A Simple Feature

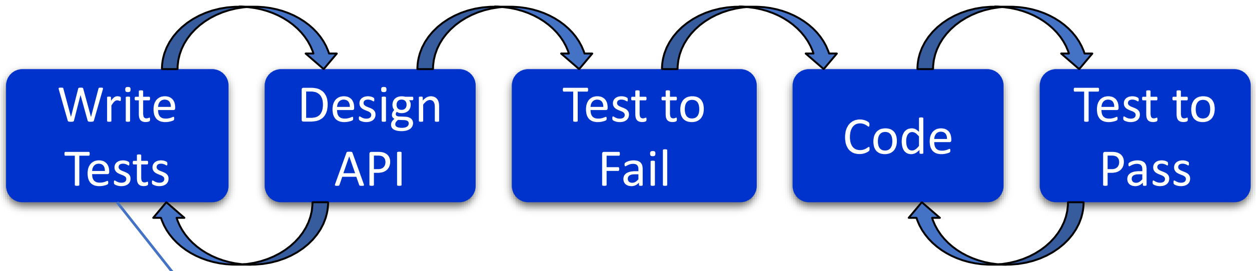


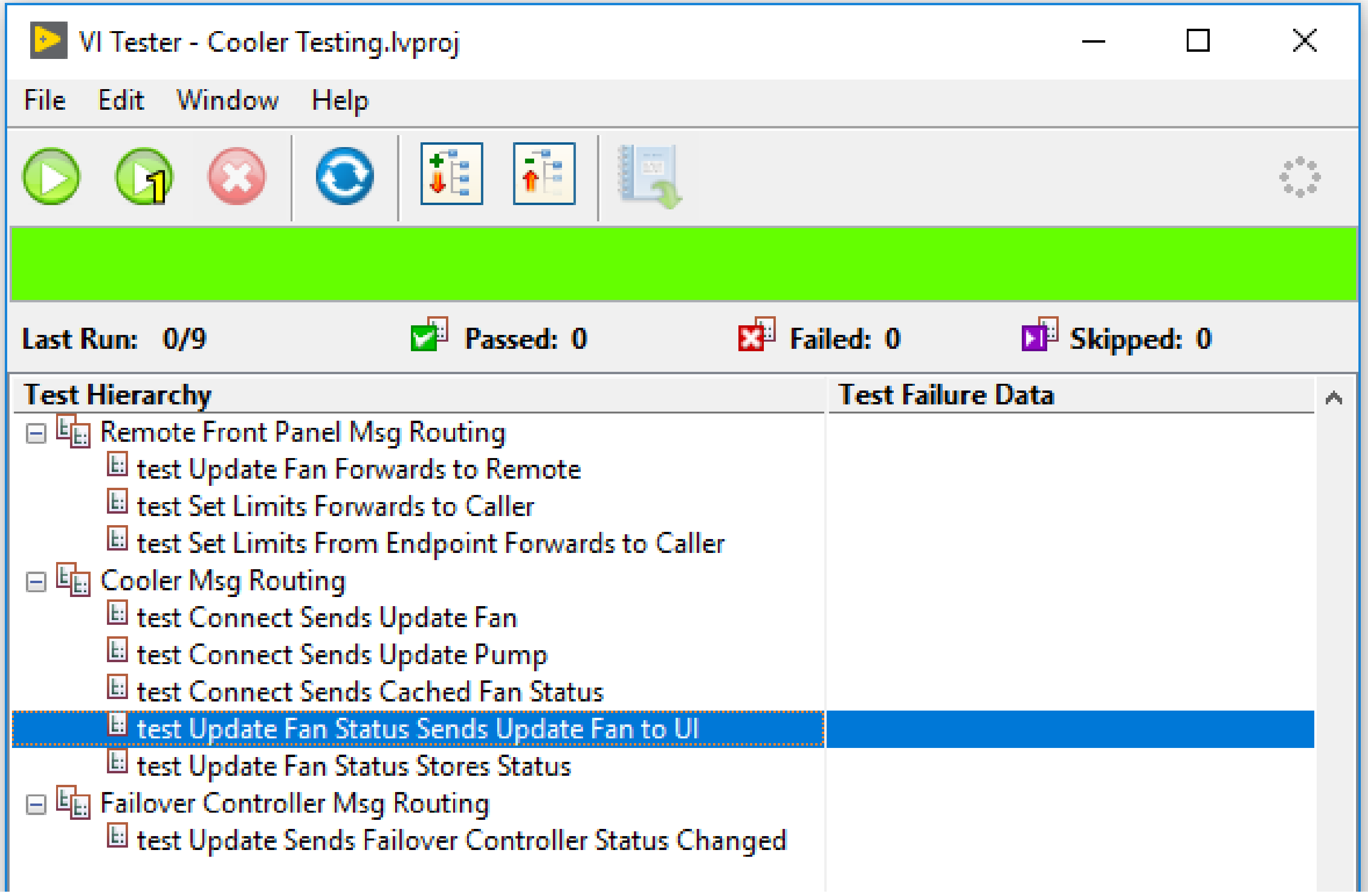
A Simple Feature, Part 1



Agenda

- Actor Framework – A Testing Challenge
- Isolating Actors
- Example – Testing a Simple Feature
  - Actor sends a message
  - Actor changes state
- A Cost Analysis

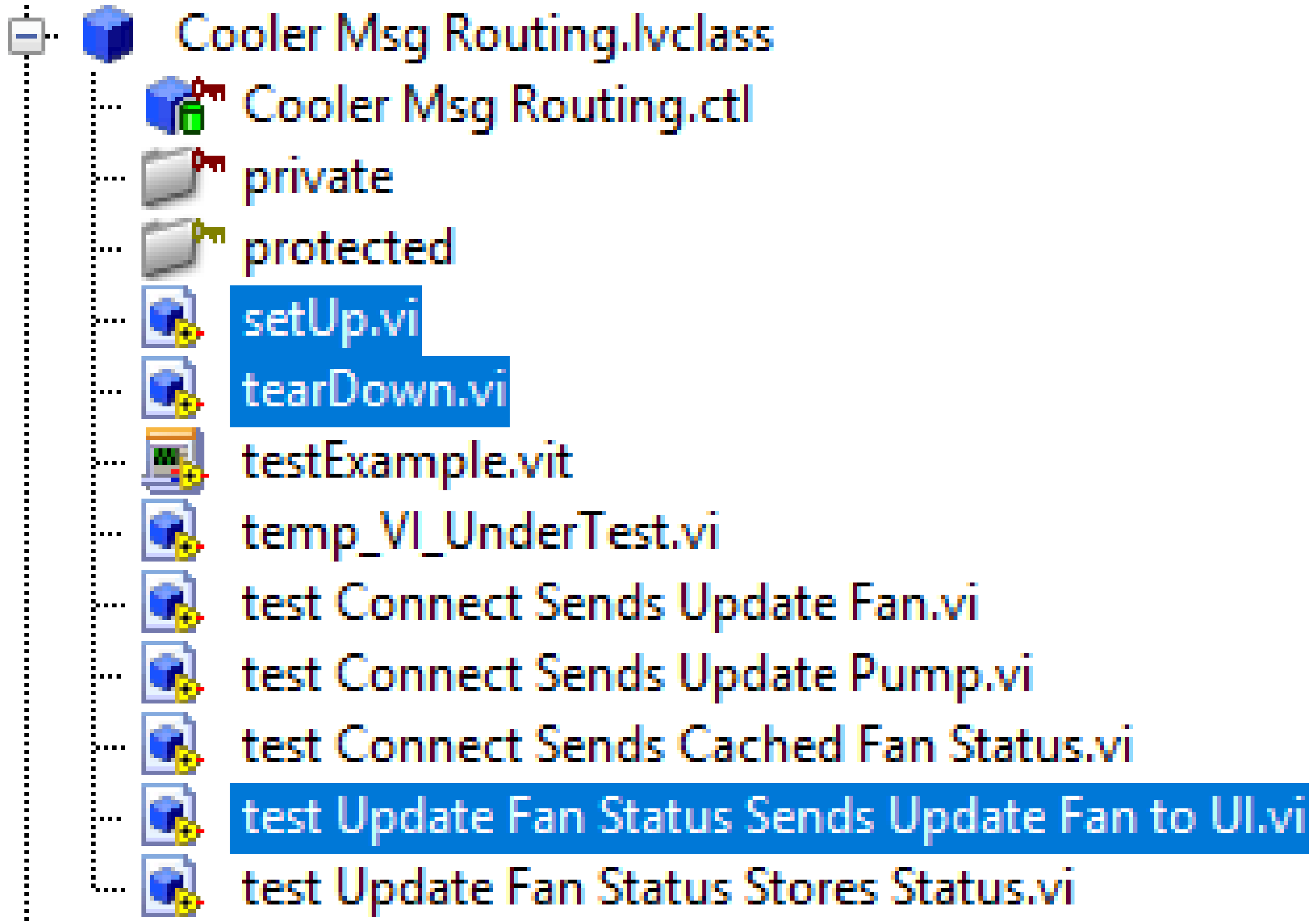




Last Run: 0/9 Passed: 0 Failed: 0 Skipped: 0

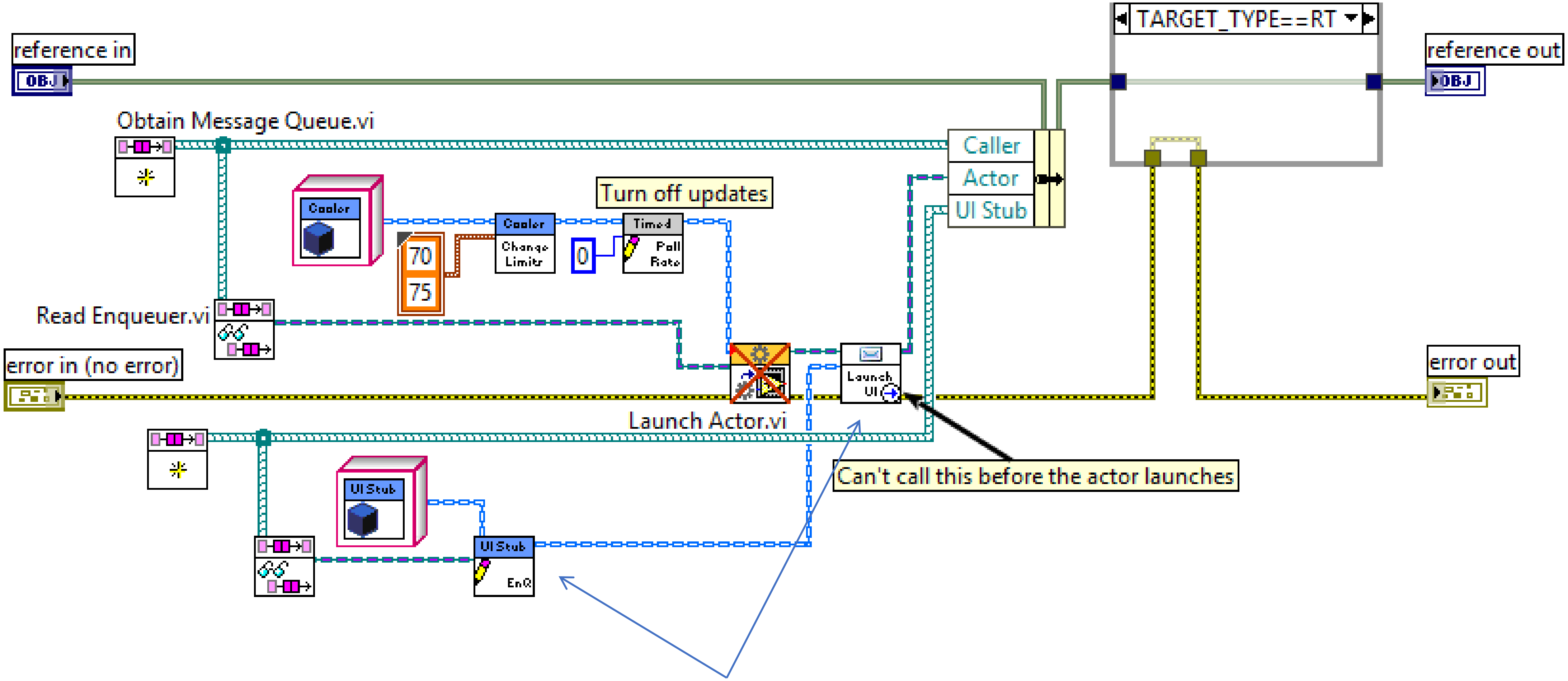
Test Hierarchy	Test Failure Data
⊟ Remote Front Panel Msg Routing	
test Update Fan Forwards to Remote	
test Set Limits Forwards to Caller	
test Set Limits From Endpoint Forwards to Caller	
⊟ Cooler Msg Routing	
test Connect Sends Update Fan	
test Connect Sends Update Pump	
test Connect Sends Cached Fan Status	
test Update Fan Status Sends Update Fan to UI	
test Update Fan Status Stores Status	
⊟ Failover Controller Msg Routing	
test Update Sends Failover Controller Status Changed	

A VI Tester Test Case



- Cooler Msg Routing.lvclass
  - Cooler Msg Routing.ctl
  - private
  - protected
  - setUp.vi
  - tearDown.vi
  - testExample.vit
  - temp_VI_UnderTest.vi
  - test Connect Sends Update Fan.vi
  - test Connect Sends Update Pump.vi
  - test Connect Sends Cached Fan Status.vi
  - test Update Fan Status Sends Update Fan to UI.vi
  - test Update Fan Status Stores Status.vi

Setup

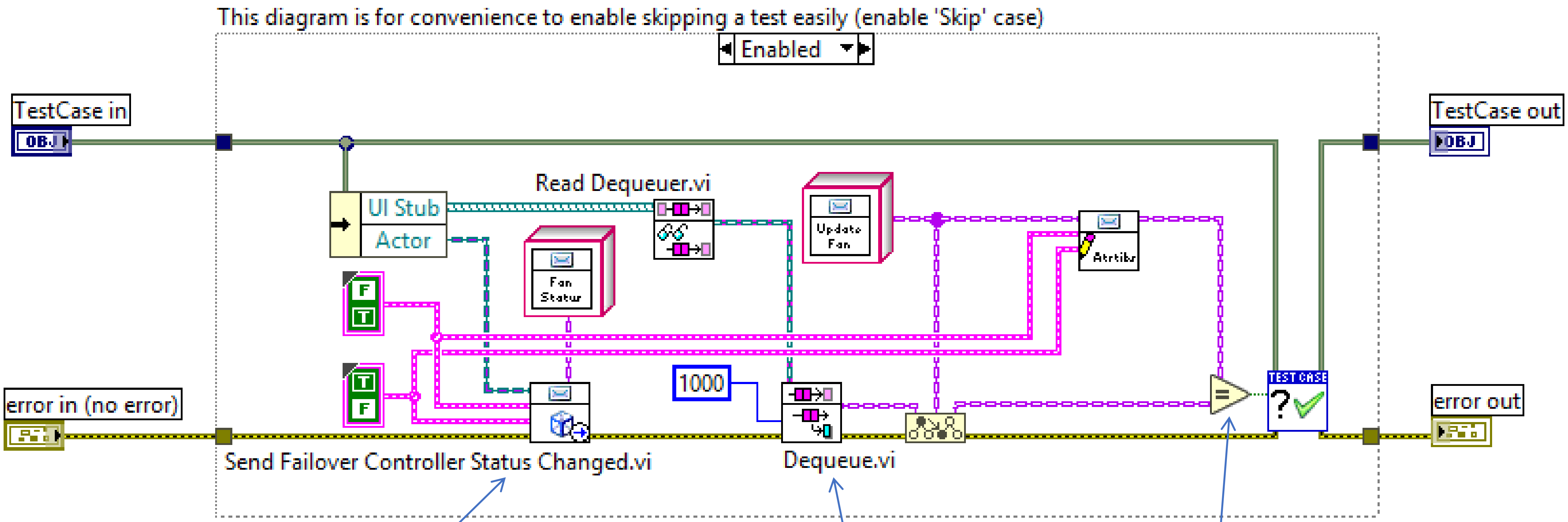


Turn off updates

Can't call this before the actor launches

Dependency injection

The Unit Test

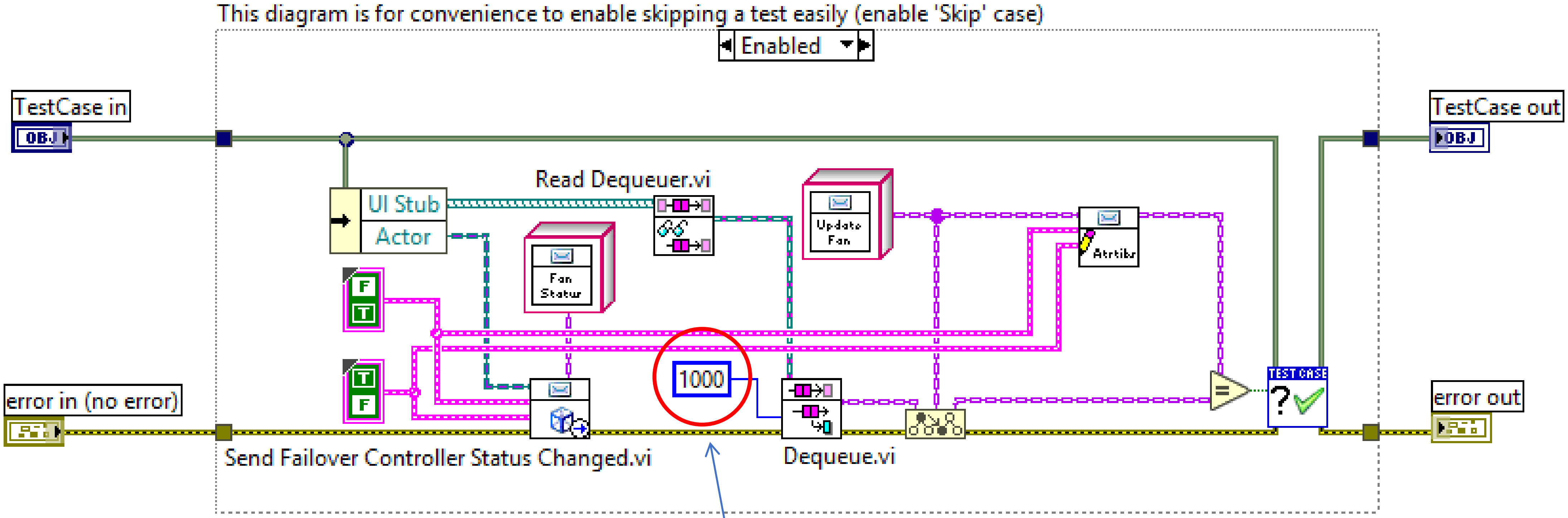


Send the (abstract) message

Receive the response

Evaluate

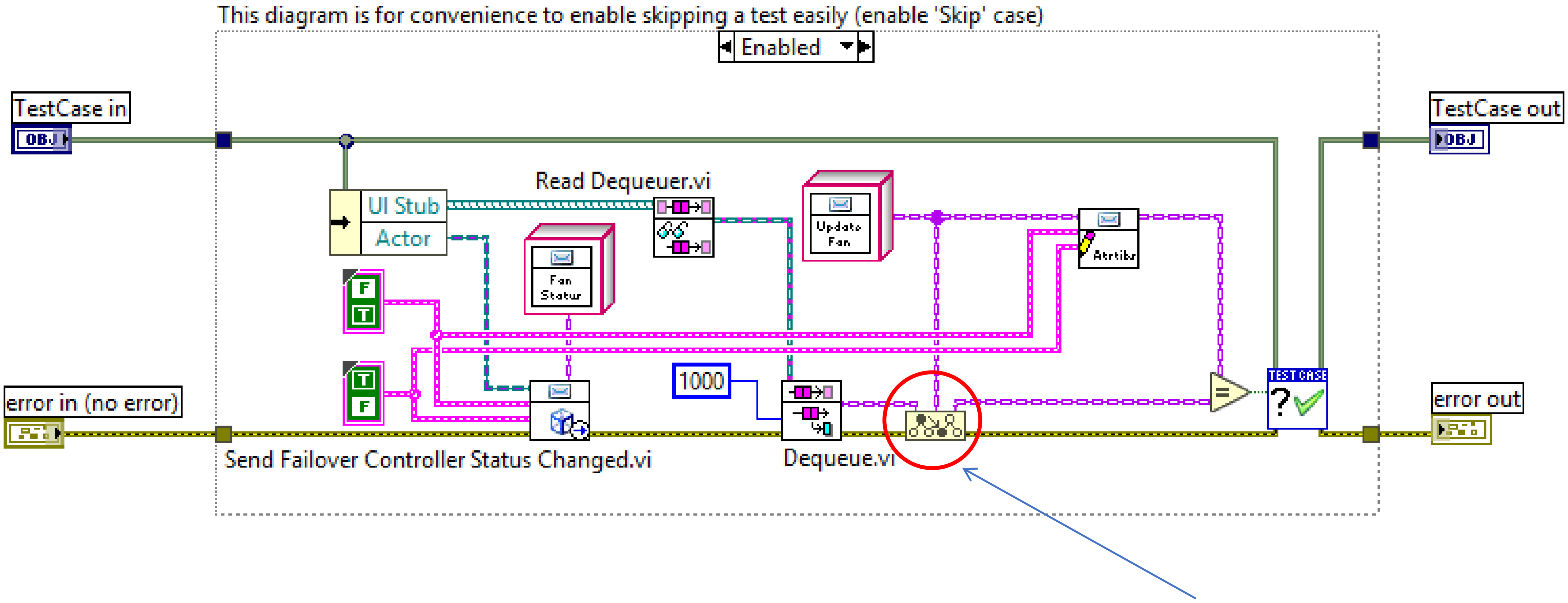
Tip: Dequeue Timeout



This diagram is for convenience to enable skipping a test easily (enable 'Skip' case)

Set a timeout here!

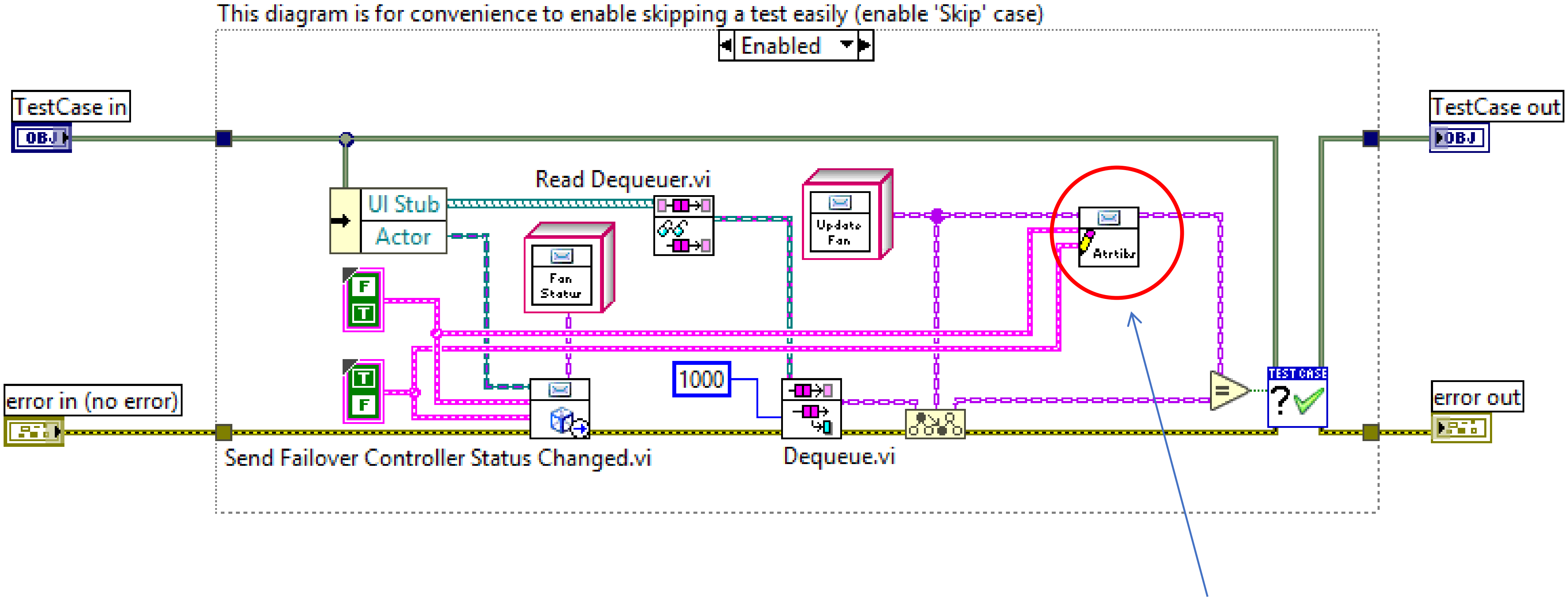
Tip: Cast to More Specific



This diagram is for convenience to enable skipping a test easily (enable 'Skip' case)

Lets you inspect return values!

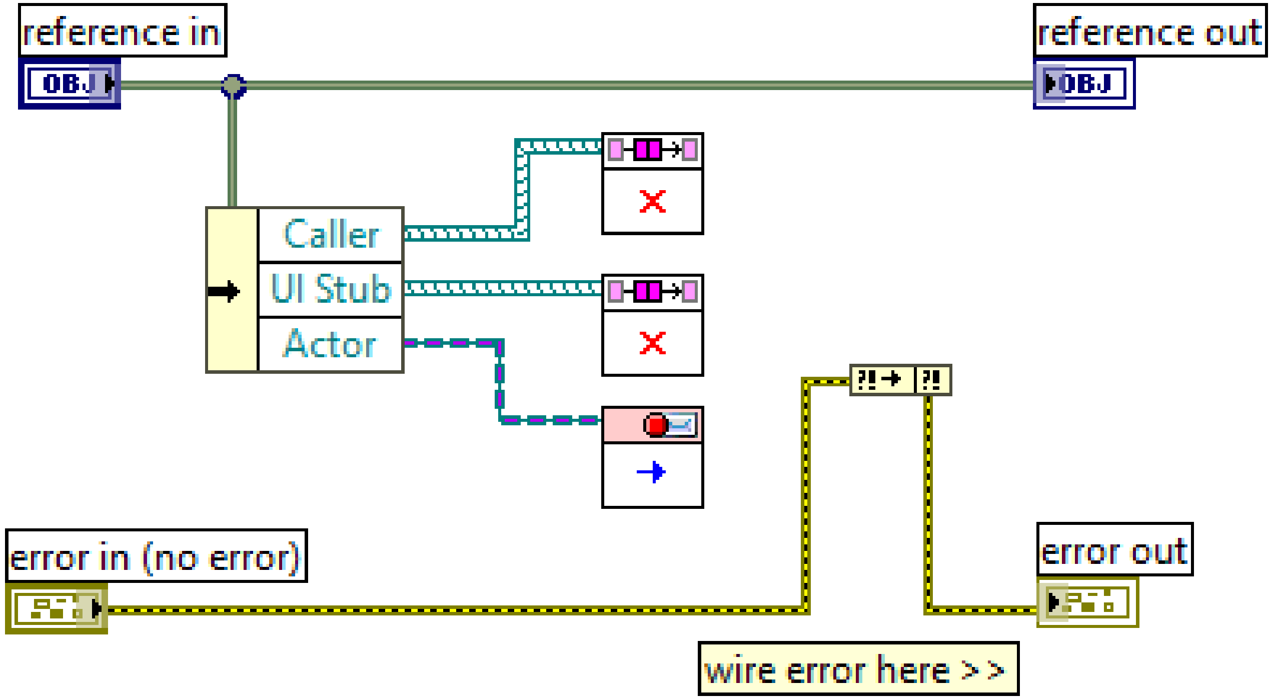
Tip: Write Accessors on Messages

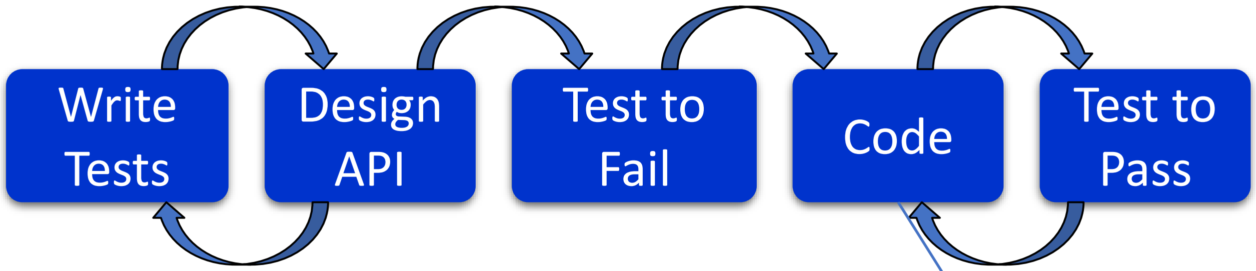


This diagram is for convenience to enable skipping a test easily (enable 'Skip' case)

Necessary to set up the compare

Test Teardown





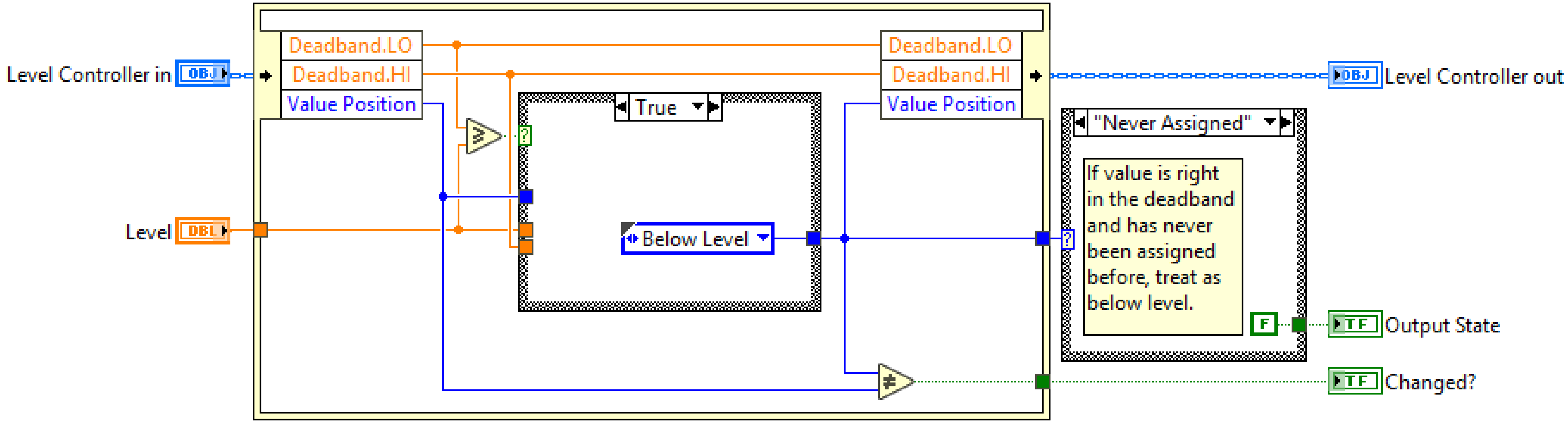
Write Tests

Design API

Test to Fail

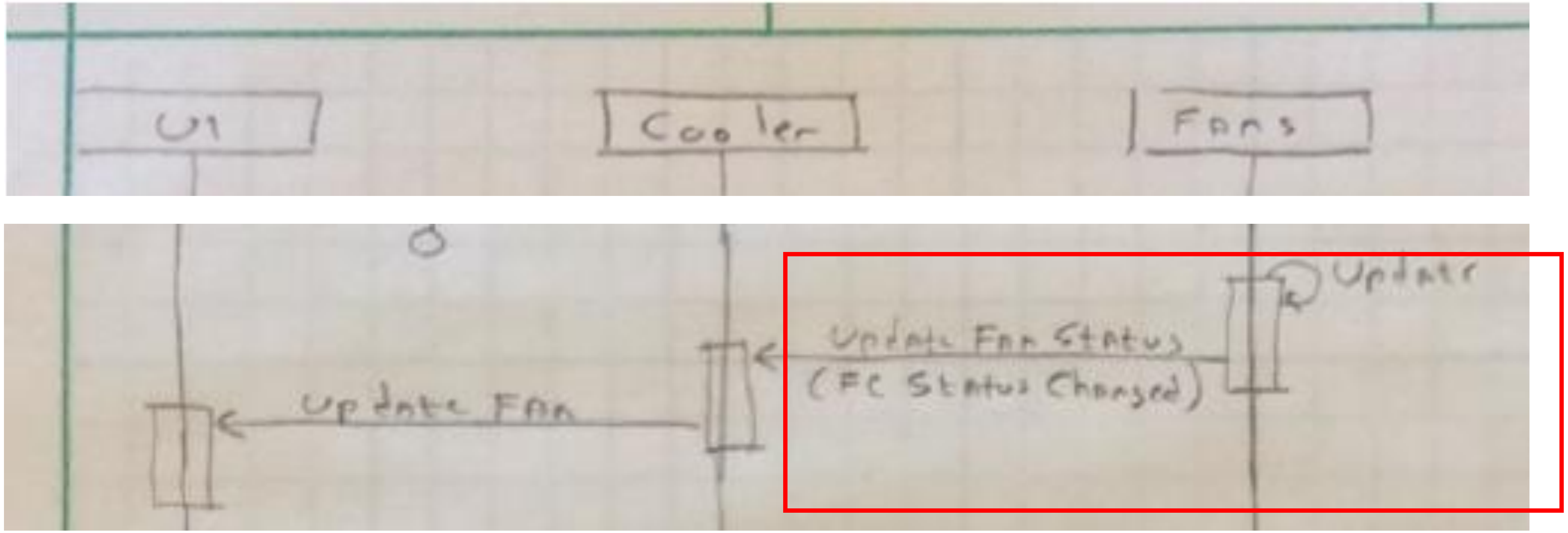
Code

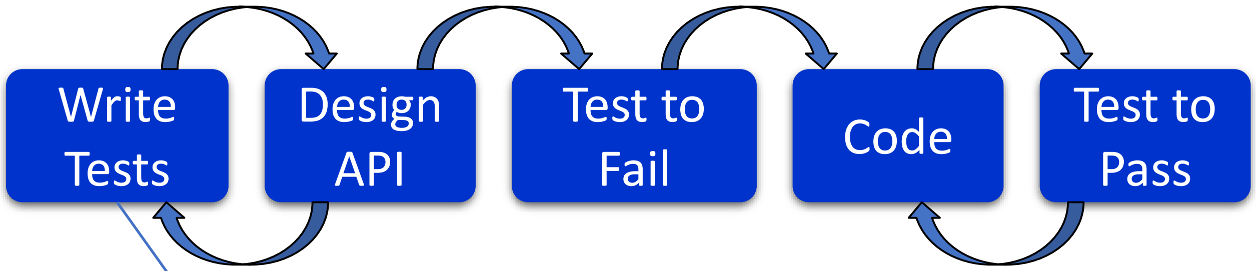
Test to Pass

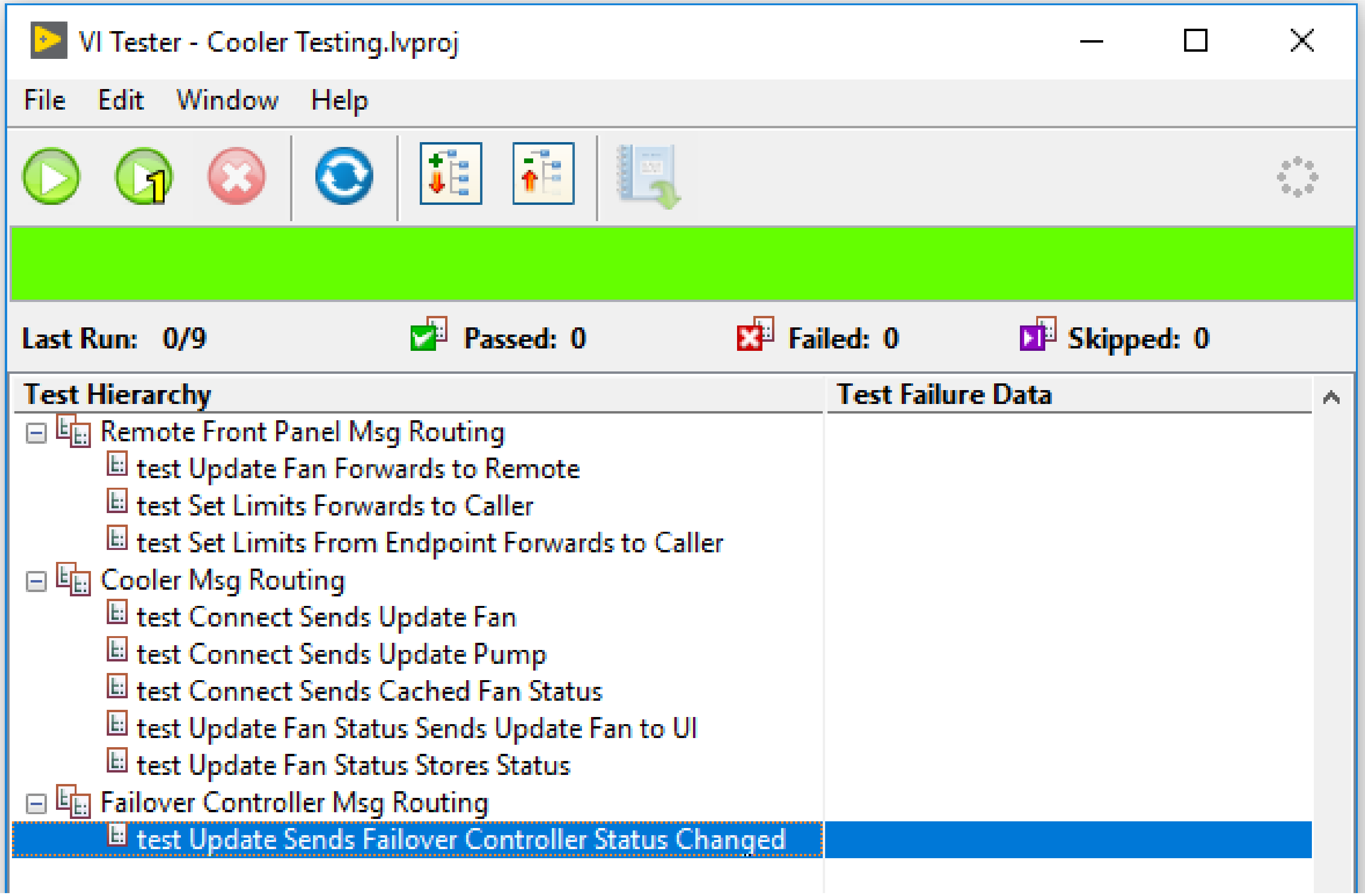


(Almost boring, in most cases)

A Simple Feature, Part 2



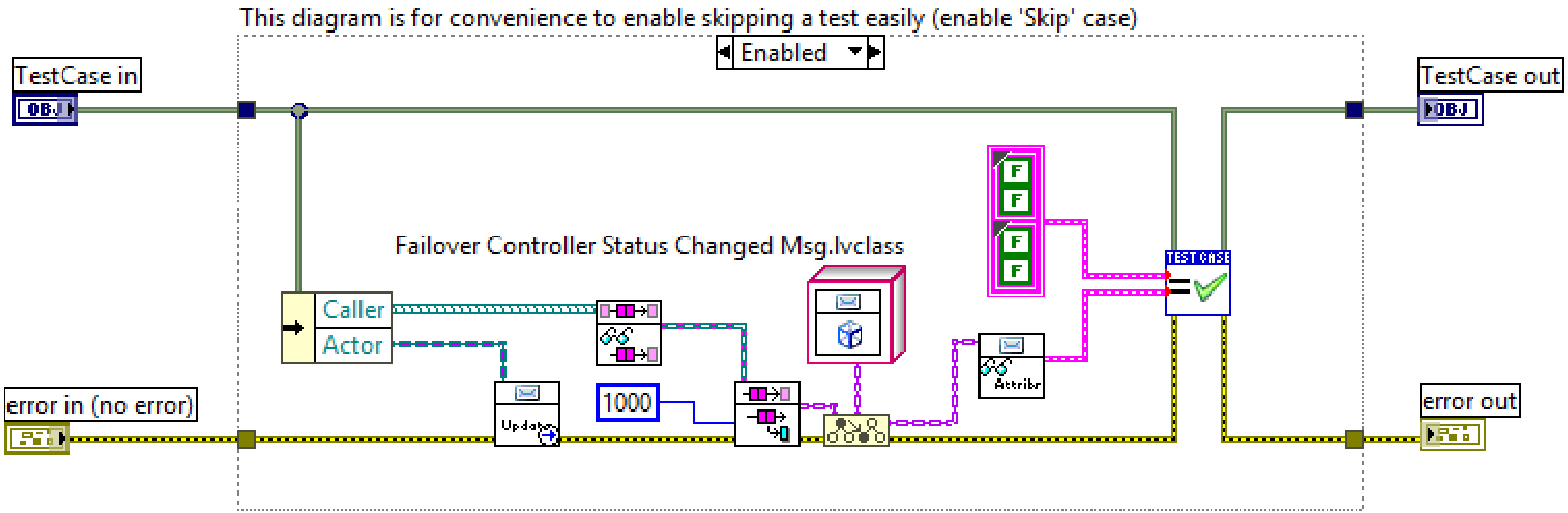




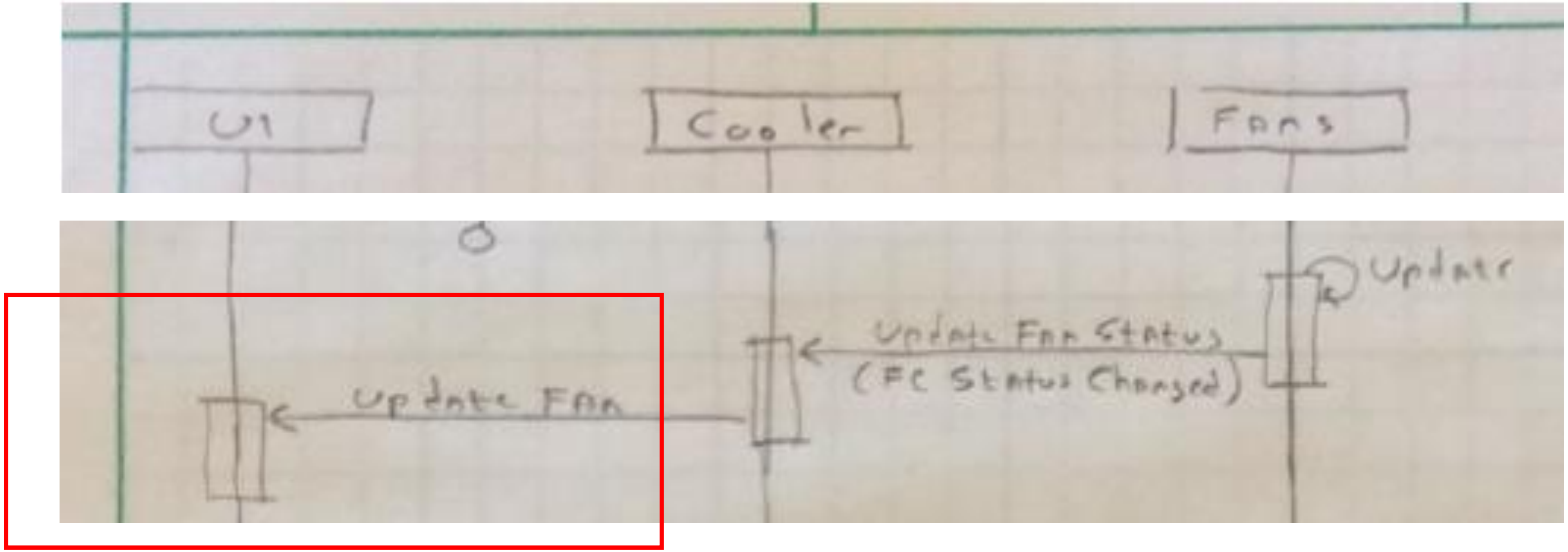
Last Run: 0/9 ✓ Passed: 0 ☒ Failed: 0 Skipped: 0

Test Hierarchy	Test Failure Data
⊟ Remote Front Panel Msg Routing	
test Update Fan Forwards to Remote	
test Set Limits Forwards to Caller	
test Set Limits From Endpoint Forwards to Caller	
⊟ Cooler Msg Routing	
test Connect Sends Update Fan	
test Connect Sends Update Pump	
test Connect Sends Cached Fan Status	
test Update Fan Status Sends Update Fan to UI	
test Update Fan Status Stores Status	
⊟ Failover Controller Msg Routing	
test Update Sends Failover Controller Status Changed	

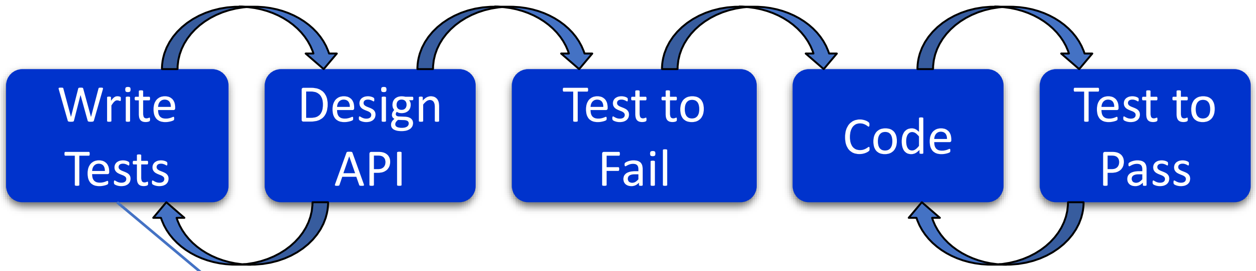
Receiving an Abstract Message


This diagram is for convenience to enable skipping a test easily (enable 'Skip' case)
Enabled
TestCase in
OBJ
TestCase out
OBJ
Failover Controller Status Changed Msg.lvclass
Caller
Actor
1000
Update
Attribs
F
F
F
F
TEST CASE
error in (no error)
error out

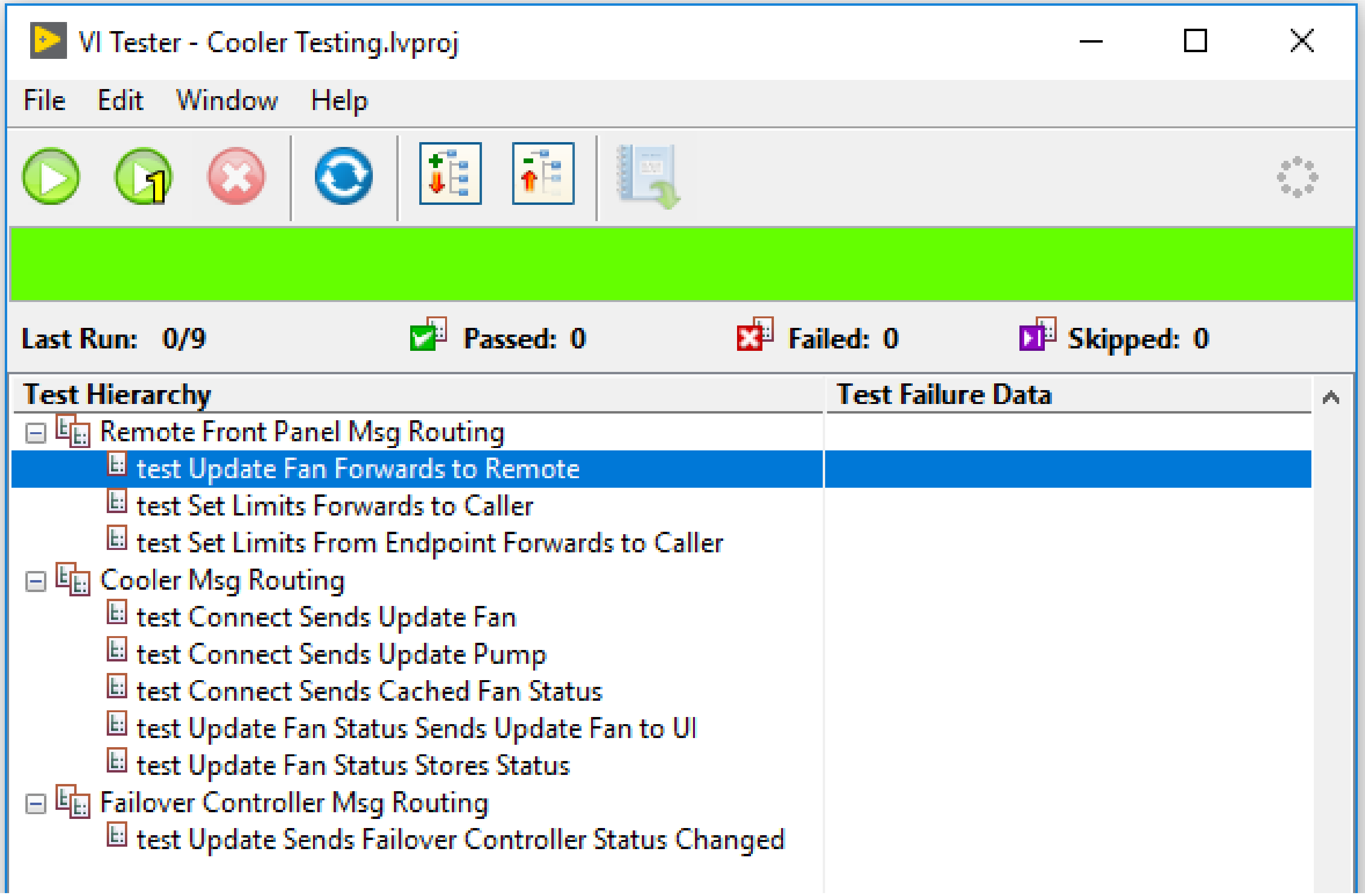
A Simple Feature, Part 3



Forward message to remote



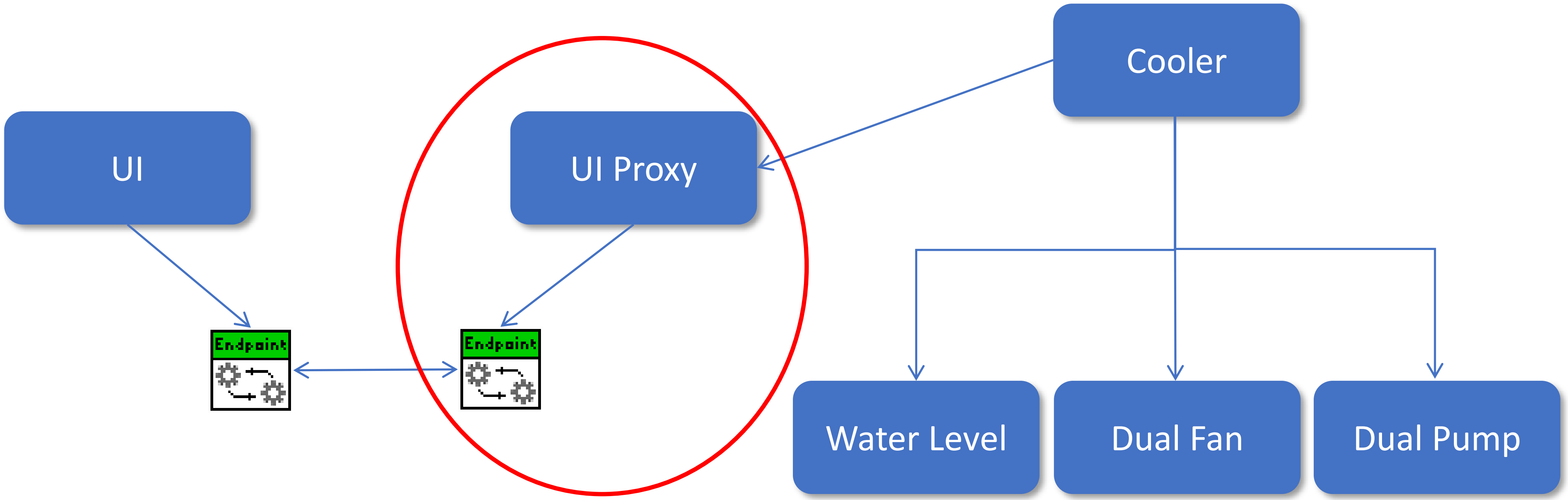
Write Tests → Design API → Test to Fail → Code → Test to Pass



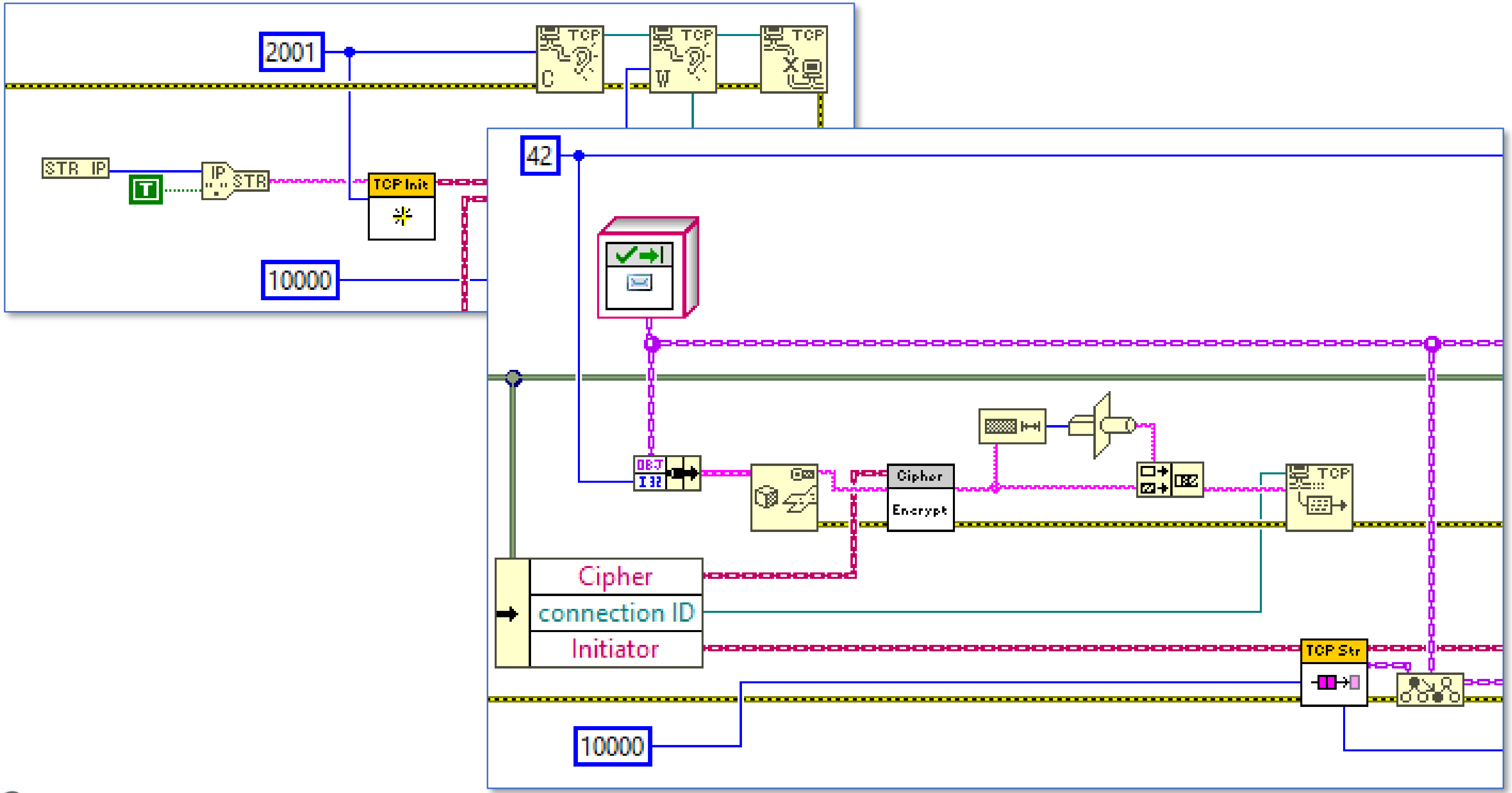
Last Run: 0/9 Passed: 0 Failed: 0 Skipped: 0

Test Hierarchy	Test Failure Data
⊟ Remote Front Panel Msg Routing	
test Update Fan Forwards to Remote	
test Set Limits Forwards to Caller	
test Set Limits From Endpoint Forwards to Caller	
⊟ Cooler Msg Routing	
test Connect Sends Update Fan	
test Connect Sends Update Pump	
test Connect Sends Cached Fan Status	
test Update Fan Status Sends Update Fan to UI	
test Update Fan Status Stores Status	
⊟ Failover Controller Msg Routing	
test Update Sends Failover Controller Status Changed	

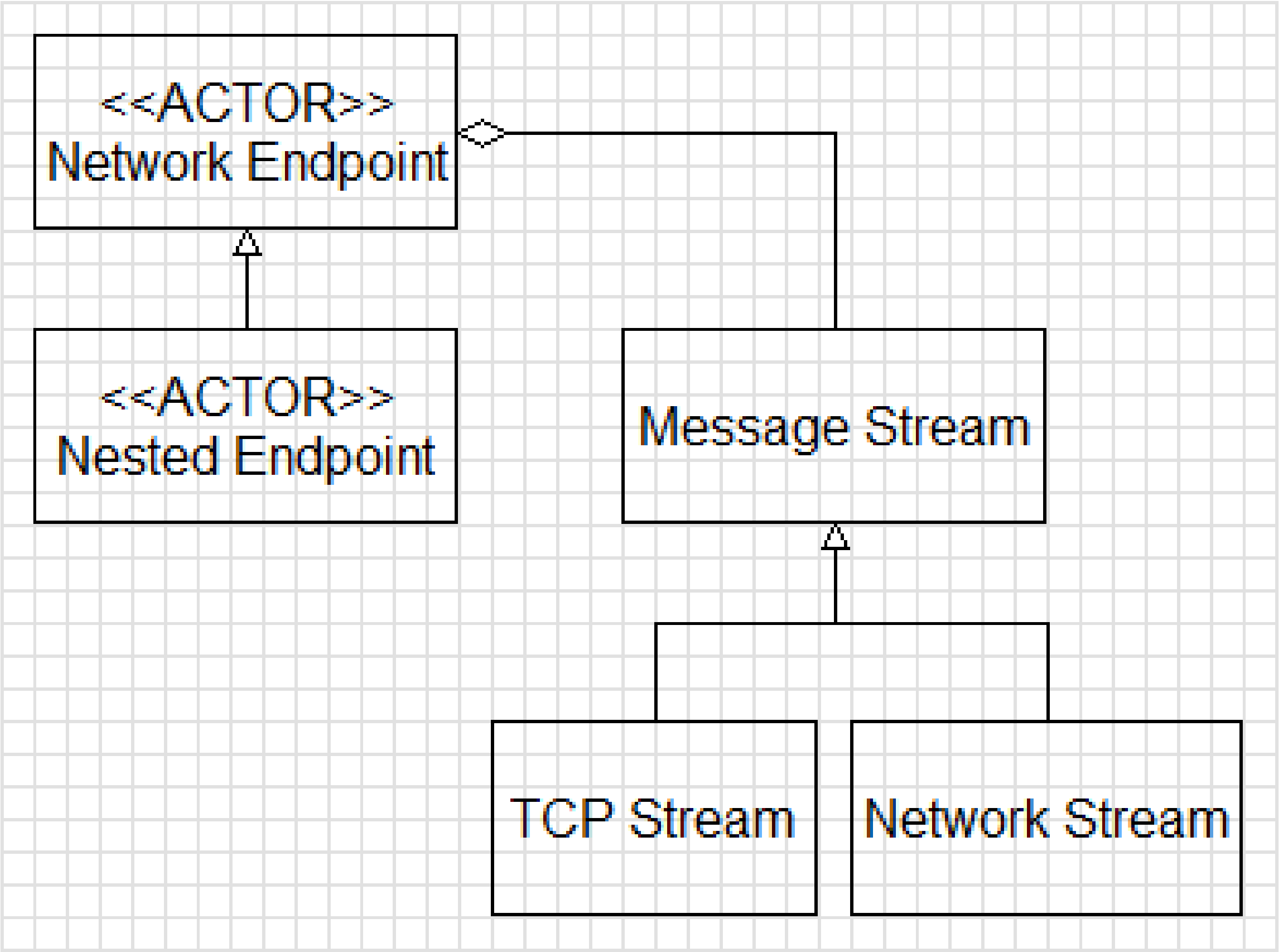
System Diagram



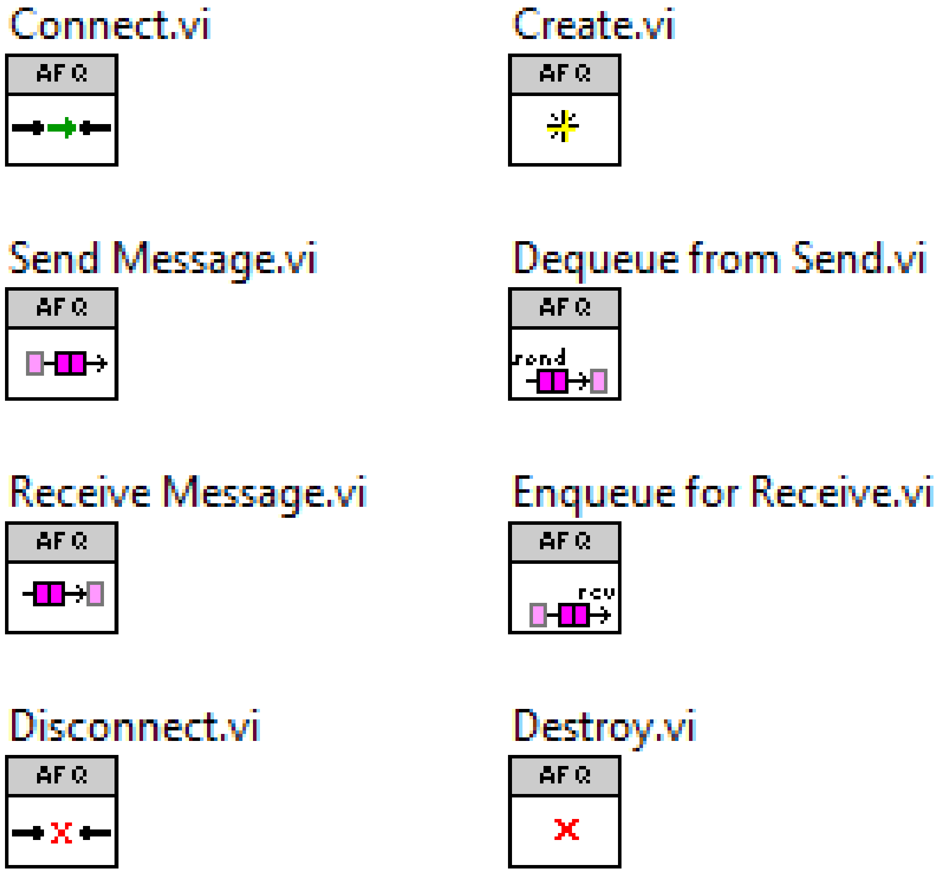
Testing Network Connectivity



Dependency Injection



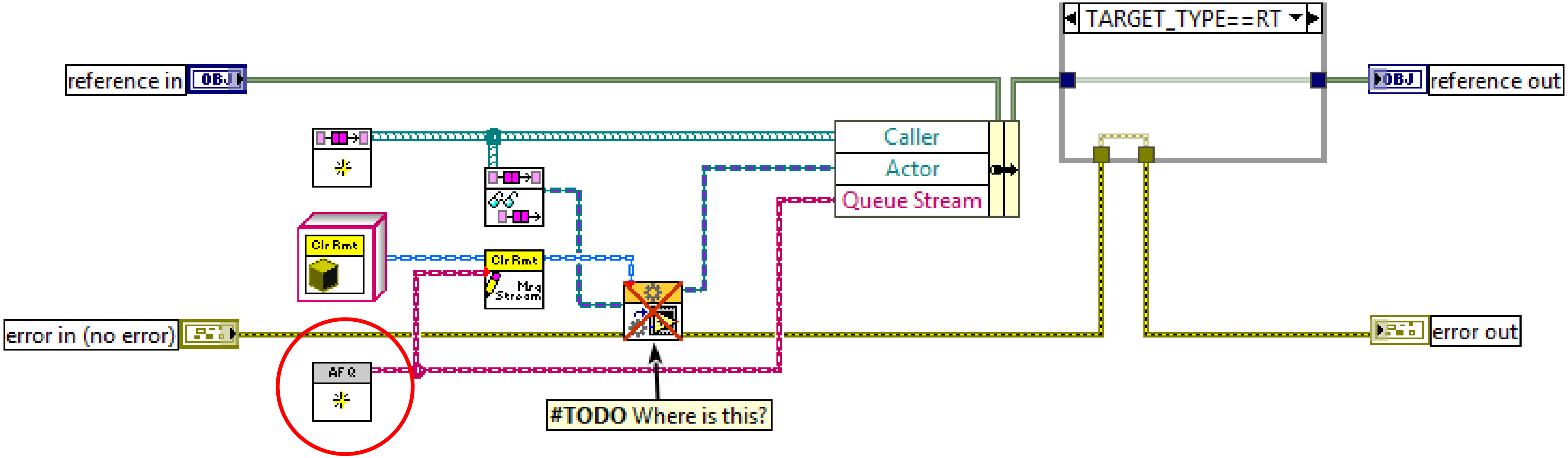
AF Queue Stream



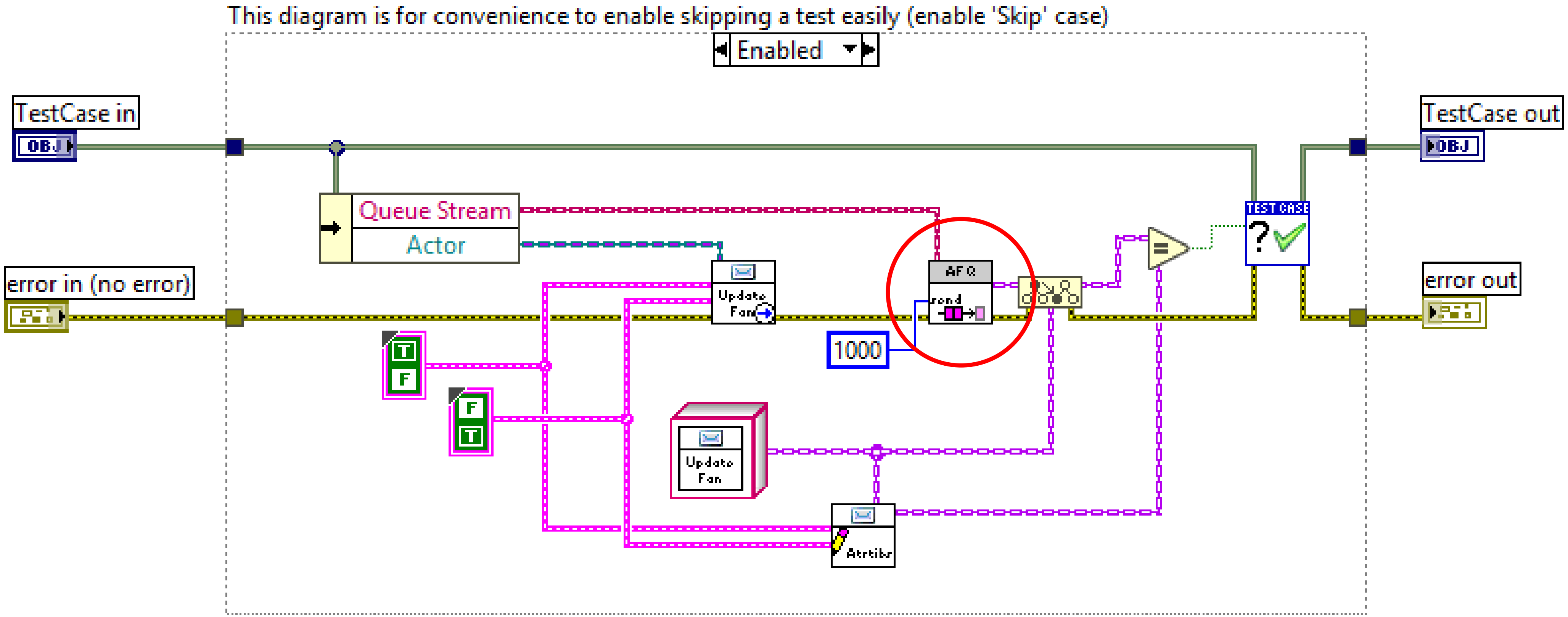
Class Overrides

Extensions for Testing

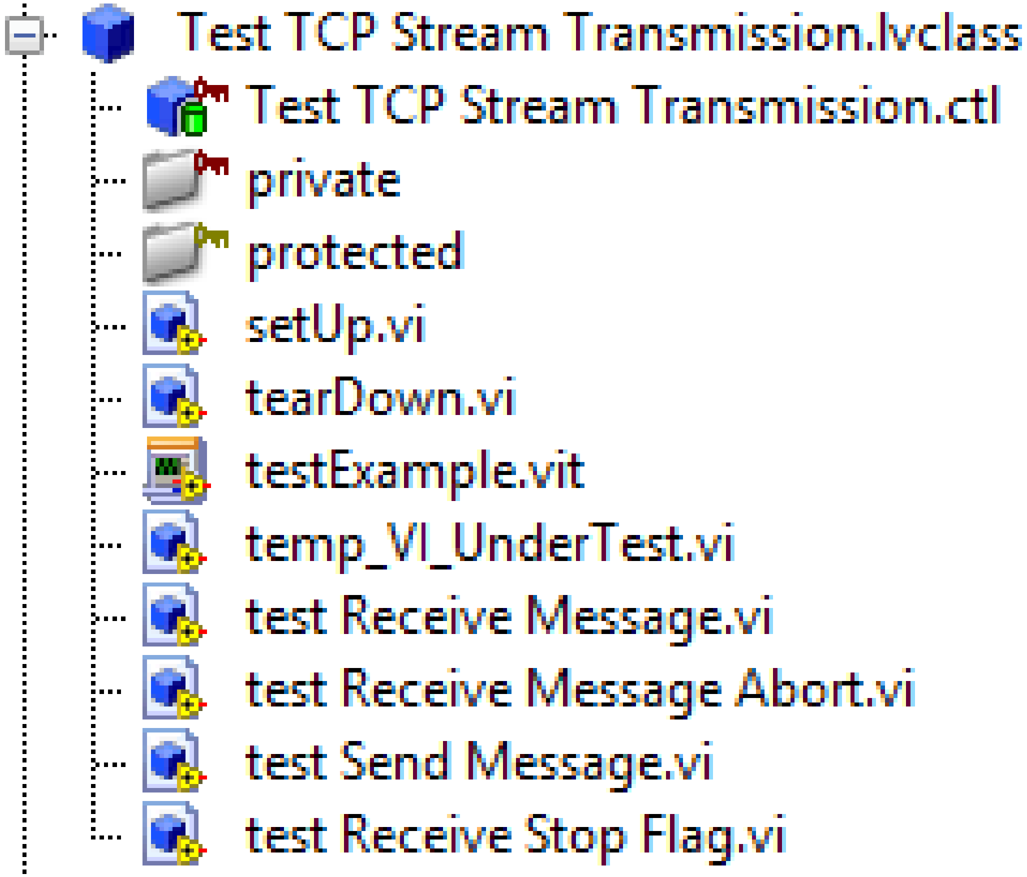
Test Setup


TARGET_TYPE==RT
reference in
OBJ
reference out
Caller
Actor
Queue Stream
Clr Rmt
Clr Rmt
Msg Stream
error in (no error)
error out
AF Q
#TODO Where is this?

A Single Test



Tip: Business Rules Belong in Classes



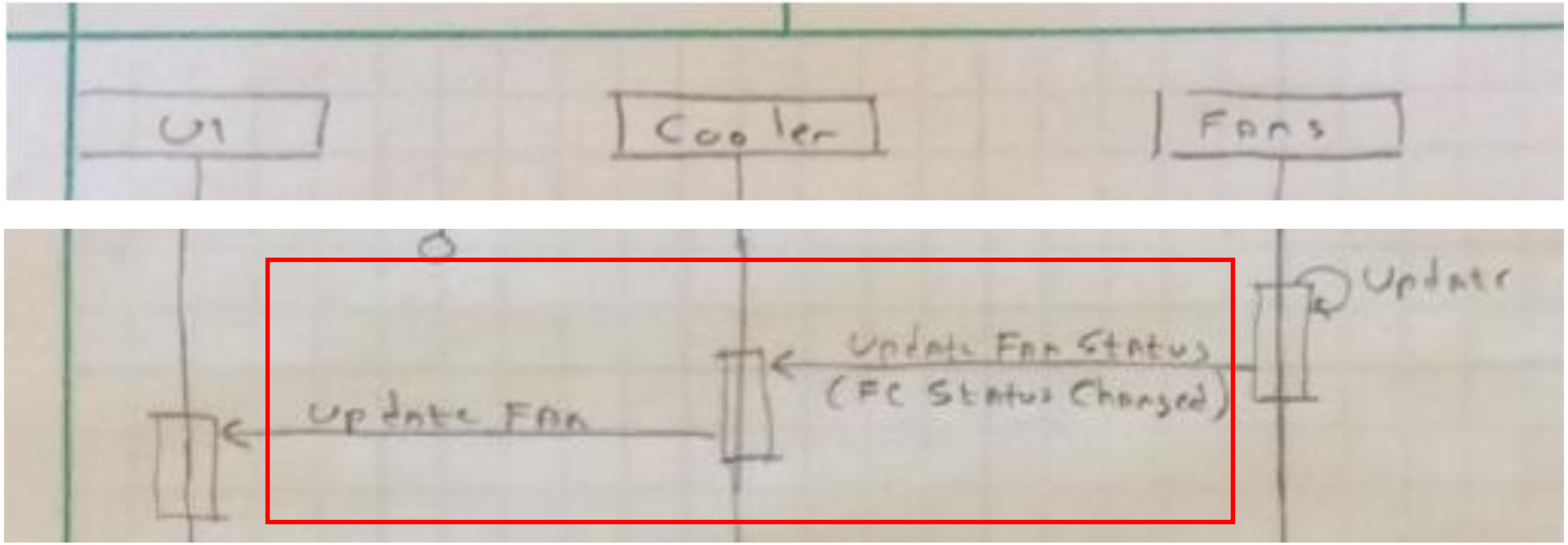
Test TCP Stream Transmission.lvclass
- Test TCP Stream Transmission.ctl
- private
- protected
- setUp.vi
- tearDown.vi
- testExample.vit
- temp_VI_UnderTest.vi
- test Receive Message.vi
- test Receive Message Abort.vi
- test Send Message.vi
- test Receive Stop Flag.vi

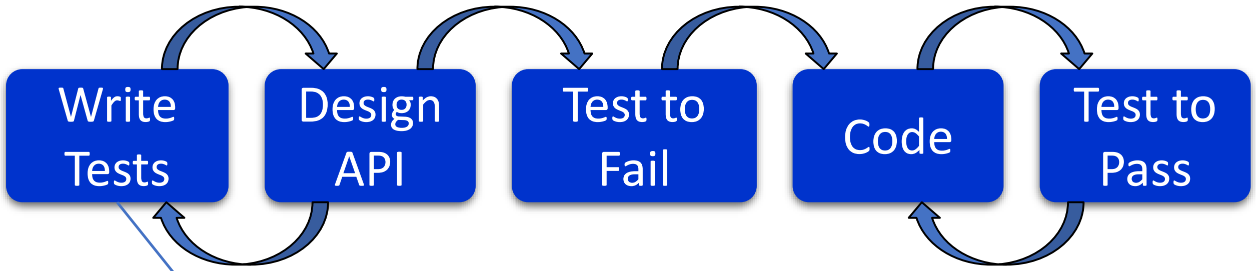
Testing is probably your most immediate and most frequently realized benefit of OO

Agenda

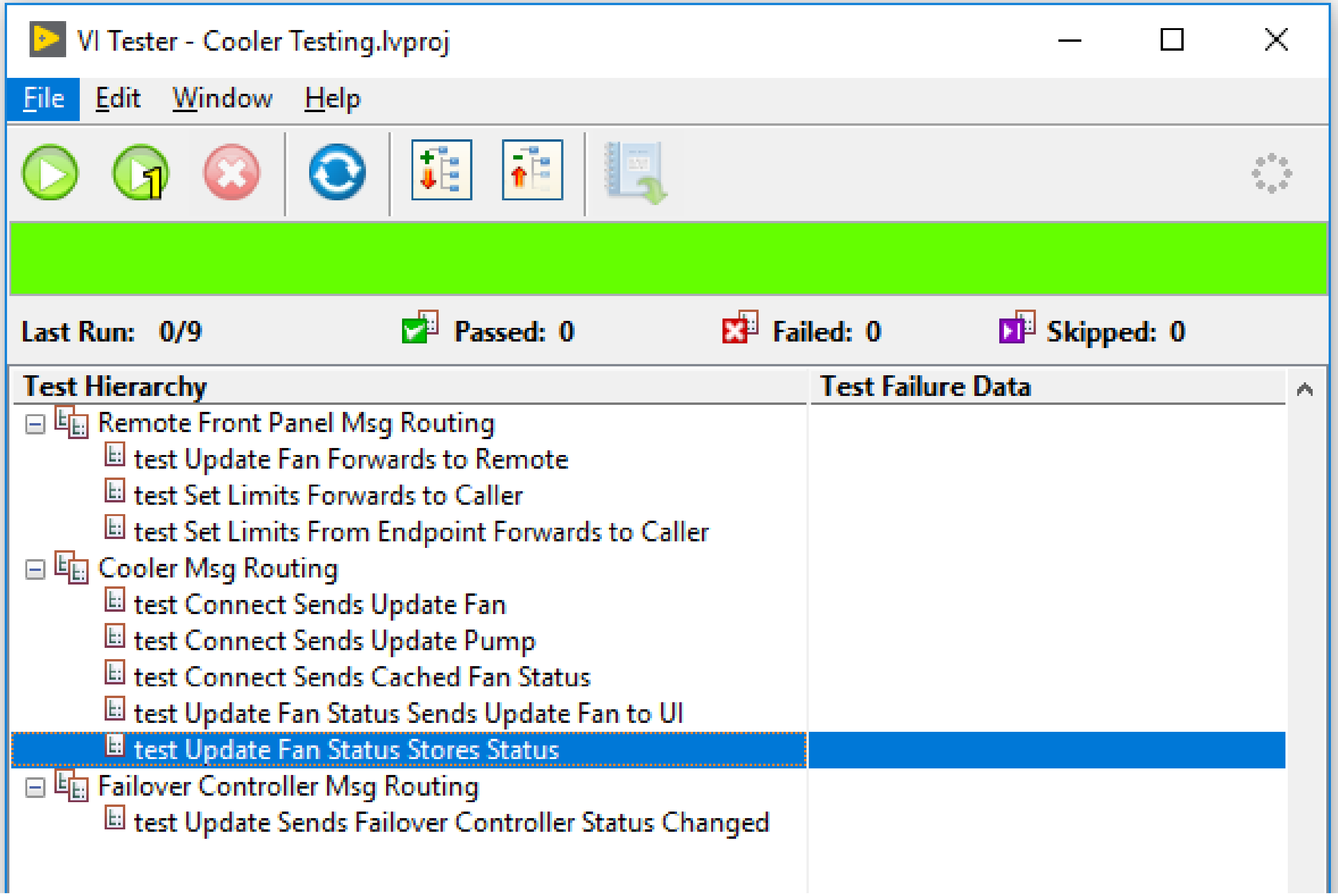
- Actor Framework – A Testing Challenge
- Isolating Actors
- Example – Testing a Simple Feature
    - Actor sends a message
    - Actor changes state
- A Cost Analysis

Actor Changes State





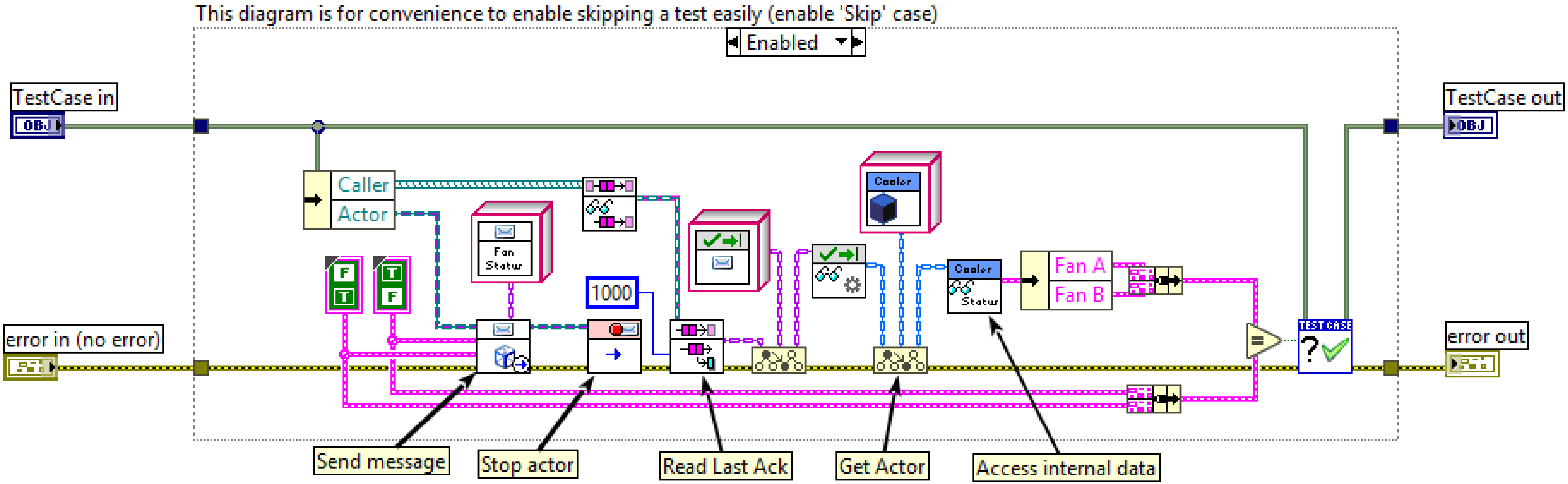
Write Tests → Design API → Test to Fail → Code → Test to Pass



Last Run: 0/9 Passed: 0 Failed: 0 Skipped: 0

Test Hierarchy	Test Failure Data
Remote Front Panel Msg Routing	
test Update Fan Forwards to Remote	
test Set Limits Forwards to Caller	
test Set Limits From Endpoint Forwards to Caller	
Cooler Msg Routing	
test Connect Sends Update Fan	
test Connect Sends Update Pump	
test Connect Sends Cached Fan Status	
test Update Fan Status Sends Update Fan to UI	
test Update Fan Status Stores Status	
Failover Controller Msg Routing	
test Update Sends Failover Controller Status Changed	

Testing Internal State



Agenda

- Actor Framework – A Testing Challenge
- Isolating Actors
- Example – Testing a Simple Feature
  - Actor sends a message
  - Actor changes state
- A Cost Analysis

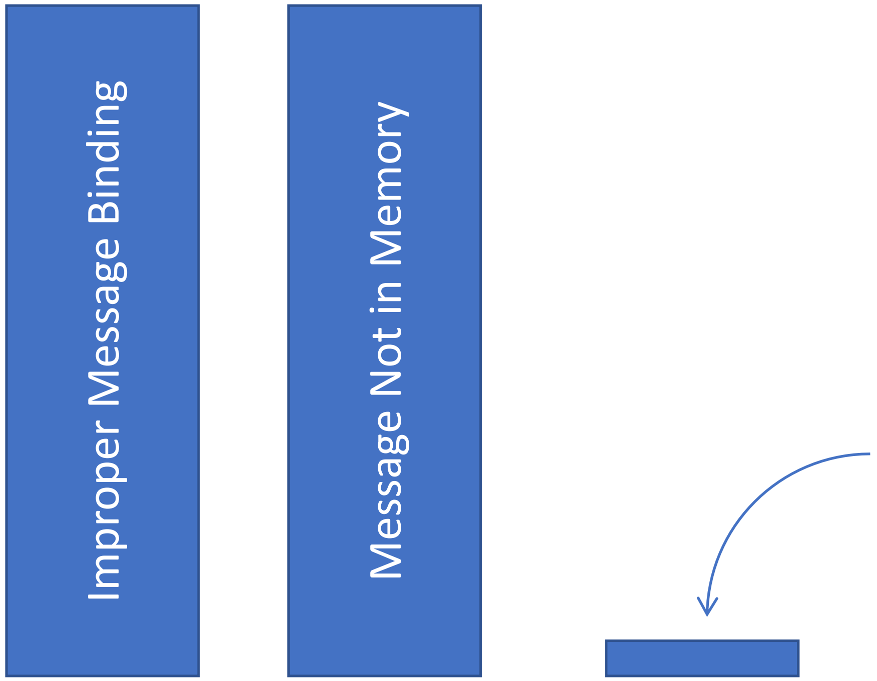
How Much Time Does it Cost?

- Activities associated with Test Driven Development:

- Writing a specification (sequence diagrams, etc.)
- Writing APIs for actor features
  - Tested actor’s method and message
  - Receiver’s method and message
- Coding the unit test
- Filling in the actor methods (tested actor)

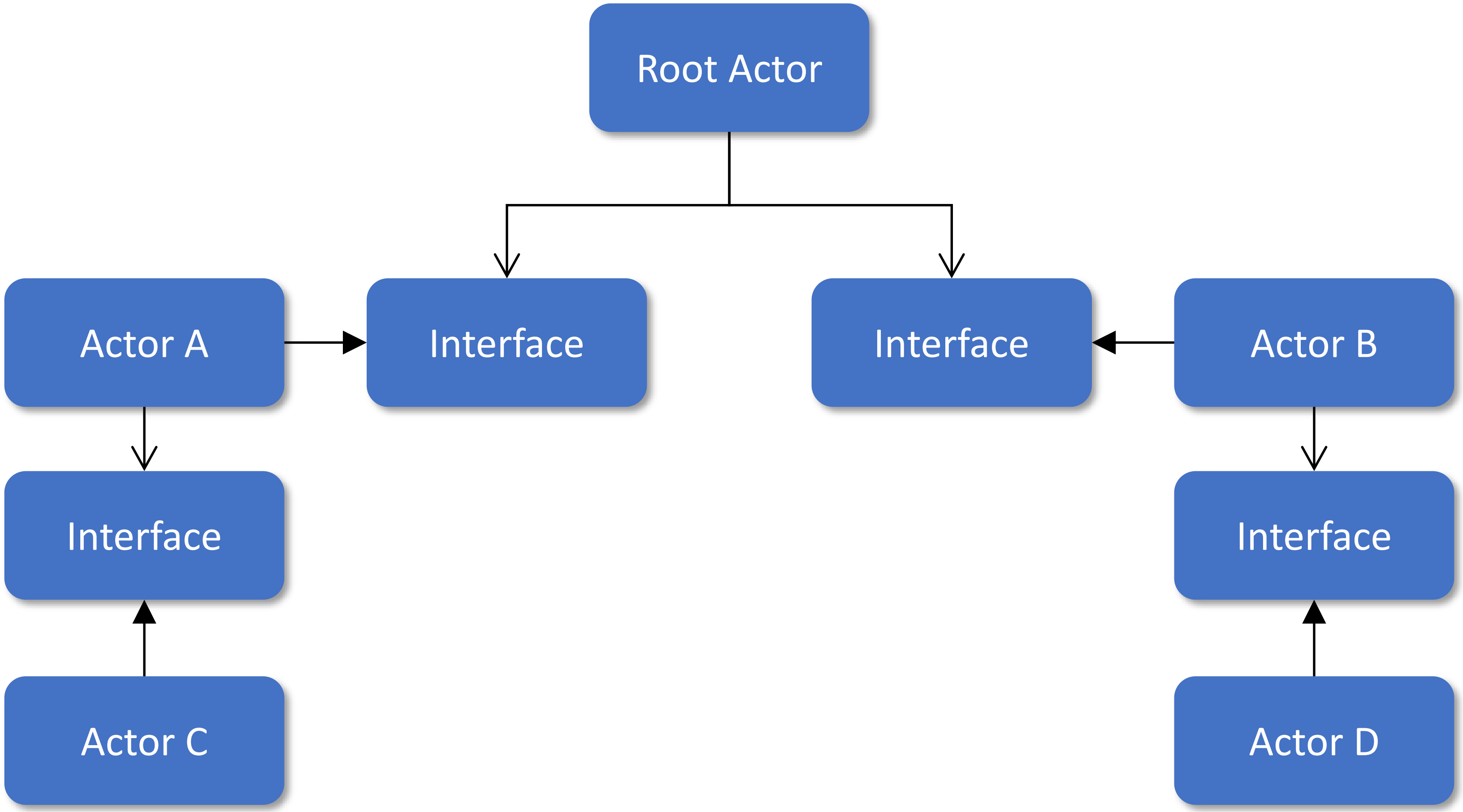
How Much Time Does It Save?

- Regression testing becomes trivial
- Bugs not discovered until integration:

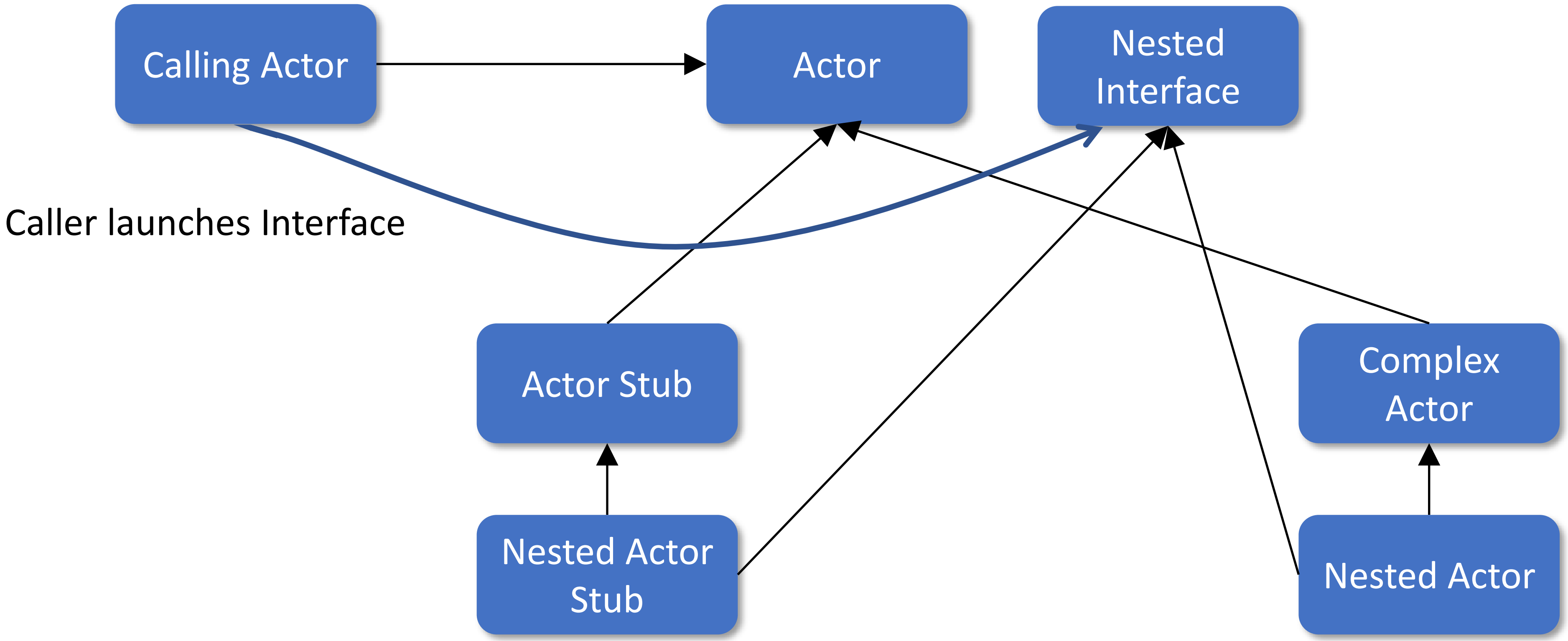


Unknown requirement
Uncommunicated assumption

Closing Thought - Design Implications



Closing Thought - Design Implications



Questions?

JUSTACS.COM

Follow me!

- LinkedIn:
  - Linkedin.com/in/allencsmith
- NI Forums:
  - justACS
- Vimeo.com/channels/ReadyOnTheWire